



Escola de Camins

Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

Desarrollo del módulo de cálculo de líneas de influencia de un programa de cálculo matricial de estructuras

Treball realitzat per:

Cristian Pujol Hernández

Dirigit per:

Joan Baiges Aznar

Luis Miguel Cervera Ruiz

Grau en:

Enginyeria Civil

Barcelona, 23/09/2015

Departament de Resistència de Materials i Estructures a

l'Enginyeria

TREBALL FINAL DE GRAU

Resumen

El presente trabajo está basado en el diseño y la programación de un nuevo módulo del programa CEM 2.0 capaz de calcular las líneas de influencia de cualquier viga horizontal con cualquier condición de contorno, véase empotramientos, apoyos, extremos libres o rótulas, usando Matlab. El objetivo principal es desarrollar una herramienta útil, práctica y fiable para complementar la docencia en el ámbito de la Resistencia de los Materiales de la Escuela Técnica Superior de Caminos, Canales y Puertos de Barcelona.

El software se ha modificado y diseñado con una interfaz de usuario simple y sencilla con lo que el usuario simplemente ha de dibujar la estructura de la cual quiere las líneas de influencia, seleccionar el botón de influencia y calcularlas. En el trabajo se explica paso a paso la programación del código y cómo Matlab lee el mismo para poder entender en todo momento cómo se han llegado a calcular las líneas de influencia.

Las soluciones finales se dan gráficamente en la interfaz de usuario de la ventana del programa CEM 2.0. El usuario puede calcular las líneas de influencia de las reacciones de todos los apoyos, las líneas de influencia de los cortantes y las líneas de influencia de los momentos flectores de una sección dada.

Para comprobar que los resultados obtenidos son correctos han sido comparados con algunos ejemplos de los power points de las clases de Estructuras y de otros libros.

Para concluir, podemos decir que se ha conseguido una herramienta útil que puede ser perfectamente utilizada para futuros estudiantes de la ingeniería interesados en el cálculo de estructuras aunque no tengan nociones de Matlab.

Índice

Resumen.....	1
Índice.....	2
Introducción y objetivos.....	4
CAPÍTULO 1. Manual del usuario CEM 2.0 – Líneas de influencia	6
1. Explicación líneas de influencia.....	7
2. Iniciación del módulo de líneas de influencia	8
2.1. Definición del entorno gráfico.....	8
3. Activación cálculo de influencia	10
3.1. Preparación de datos y creación de nuevas variables	11
3.2. Cálculo y creación de las líneas de influencia	11
3.2.1. Cálculo del momento flector, cortante y reacciones	12
3.2.2. Función principal: dibujo de los diagramas de influencia	14
3.2.3. Colocación, escala y coloreado de los diagramas de influencia.....	16
3.2.4. Activación botones y visibilidad diagramas	19
3.2.5. Presentación de los resultados	19
CAPÍTULO 2. Explicación del código en MATLAB	20
1. Creación y acción del panel y botones de influencia. Definición del entorno gráfico	23
1. Función panell_botons_influencia	26
1.1. Función popupcallbackinf	26
1.1.1. Función autopopupcallbackinf.....	26
1.2. Función diagTyInf	27
1.3. Función diagMzInf.....	28
1.4. Función visibilitat_diagramesInf	28
2. Redimensionamiento	28
3. Cambios en la activación de los botones de cálculo	29
4. Acción que se ejecuta con el botón calcular y construcción del diagrama de influencia	32
4.2.1. Función prepara_dades_calculinfluencia.....	59
4.2.2. Función diagramainfluencia	59

4.2.3. Función esborra_diagramesireaccionsinfluencia.....	60
4.2.4. Función calcul_tallant_moment_reaccions	60
4.2.4.1. Función calcul_esforços_influencia	61
4.2.4.2. Función dades_diagramainfluencia	62
4.2.5. Función correccio_tallant.....	62
4.2.6. Función dibuix_diagrames_reaccions_moment	65
4.2.6.1 Función dibudiag_reaccions_moment.....	66
4.2.6.2. Función dibudiag_tallant.....	66
4.2.7. Función colordiagramareaccions	66
CAPÍTULO 3	67
1. Ejemplos de líneas de influencia	68
1.1. Viga biapoyada	68
1.2. Viga biapoyada con voladizo	69
1.3. Viga con un apoyo y un empotramiento.....	70
1.4. Viga biempotrada.....	71
1.5. Viga con rótula	72
1.6. Viga isostática de 3 vanos con 2 rótulas.....	74
2. Conclusiones.....	78
3. Bibliografía	79

Introducción y objetivos

En este trabajo se ha desarrollado, mediante Matlab, un código de cálculo de líneas de influencia con interfaz gráfica. El código se ha implementado en el programa CEM 2.0 – una mejora de diferentes estudiantes de las titulaciones de Ingeniería de Caminos, Canales y Puertos, Ingeniería de Obras Públicas y grado en Ingeniería de la Construcción del programa CEM, desarrollado originalmente por el profesor Dr. Joan Baiges-.

CEM 2.0 incluye los módulos de cálculo estático, cálculo plástico, cálculo dinámico e inestabilidad. Con este trabajo se ha incluido un módulo de cálculo de líneas de influencia dentro del cálculo estático. El objetivo del programa es proporcionar una herramienta útil para complementar la docencia de los alumnos en las asignaturas de Resistencia de los materiales impartidas en la escuela.

Un factor importante para el desarrollo de este módulo ha sido que la interfaz sea sobre todo intuitiva, cómoda y fácil de usar y que se complementara adecuadamente a la interfaz anterior. De esta manera, el programa se ha desarrollado con una interfaz gráfica, tanto de introducción de datos como de salida, clara y sencilla para que los futuros alumnos no tengan que hacer uso del manual expuesto de este trabajo.

El trabajo se divide en dos grandes capítulos y un tercero donde se tiene como objetivo plasmar el trabajo desarrollado. Dichas partes, denominadas capitulos, son:

1. Elaboración del Manual del Usuario del módulo de Líneas de Influencia en CEM 2.0.
2. Descripción detallada de los códigos principales del programa y de la estructura de este.
3. Ejemplos, conclusiones y bibliografía.

La numeración de las figuras y códigos se ha llevado a cabo de forma ordenada independientemente del capítulo y para las referencias bibliográficas de la Bibliografía se ha utilizado la normativa española APA 2014.

Finalmente, el código del programa adjunto a este trabajo contiene los comentarios que se han creído convenientes para facilitar su interpretación, así como para ayudar a futuros desarrolladores y colaboradores del programa.

CAPÍTULO 1. Manual del usuario CEM 2.0

– Líneas de influencia

1. Explicación líneas de influencia

En el ámbito de las estructuras y en el programa cme2 se suelen analizar estructuras sometidas a cargas de magnitud conocida y ubicación definida. Un ejemplo de estas cargas, llamadas cargas muertas, es el peso propio. Sin embargo, existen otros estados de carga que pueden estar o no estar presentes (sobrecarga) o que pueden cambiar de ubicación (cargas móviles). Estas cargas se conocen como cargas vivas de una estructura.

Para los efectos del diseño de una estructura interesa conocer el valor máximo de los efectos que producen las cargas externas en ella. Siempre hay una posición de las cargas vivas para la cual se produce el valor máximo de un efecto. Para determinar esta posición y ese valor se usan las líneas de influencia.

La línea de influencia es una curva cuya ordenada, en este trabajo será el eje Y, muestra una variación del efecto causado por una carga unitaria que se mueve a través de la estructura. Así pues, la ordenada en cualquier punto del diagrama representa el valor del efecto causado por la carga unitaria cuando ésta está actuando en el correspondiente punto de la estructura respecto una sección de estudio.

La unidad de las ordenadas de una línea de influencia corresponde a la unidad del efecto que se cuantifica dividido por la unidad de la fuerza. Así la unidad de la ordenada de la línea de influencia de una fuerza reacción en un apoyo o de una fuerza de corte en una sección transversal es adimensional.

Las líneas de influencia quedan definidas si se conoce la forma, la escala y el signo. Durante todo el trabajo se seguirá el convenio de signos del módulo de cálculo estático.

La determinación de las líneas de influencia se hará tanto en estructuras isostáticas simples como hiperestáticas como lo son las vigas pero su aplicación se podría extender a arcos, marcos, pórticos...

Las líneas de influencia en una estructura isostática son líneas rectas aunque si la estructura no es isostática las líneas ya no serán rectas. De este modo la construcción de las líneas de influencia se reduce a determinar las ordenadas de unos cuantos puntos de ellas.

Cuando la carga móvil que actúa sobre la estructura es una carga concentrada de magnitud P , la magnitud del efecto debido a esta carga es igual al producto de la carga P por la ordenada de la línea de influencia para el efecto considerado en el punto donde se ubica la carga P . Por esa razón, la carga que se desplaza a lo largo de la estructura es unitaria.

Para obtener el máximo valor del efecto, la carga P deberá ubicarse en la posición para la cual la ordenada de la línea de influencia es máxima.

2. Iniciación del módulo de líneas de influencia

A continuación se explicará el manual del usuario del programa CEM 2.0. Ha sido explicado con un ejemplo sencillo donde podemos ver cómo Matlab lee el código para entender mejor su funcionamiento.

2.1. Definición del entorno gráfico

En este punto se explicará el funcionamiento del módulo para el cálculo de las líneas de influencia del programa cme2, paso a paso. Cabe decir que, una vez abramos el programa, saldrá directamente la pantalla de inicio definitiva (Figura 1) preparada para poder dibujar directamente la estructura, por ese motivo, en este punto se explicará tal y como Matlab ejecuta el programa y las diferentes funciones de forma ordenada para darnos cuenta de cómo está creado el módulo.

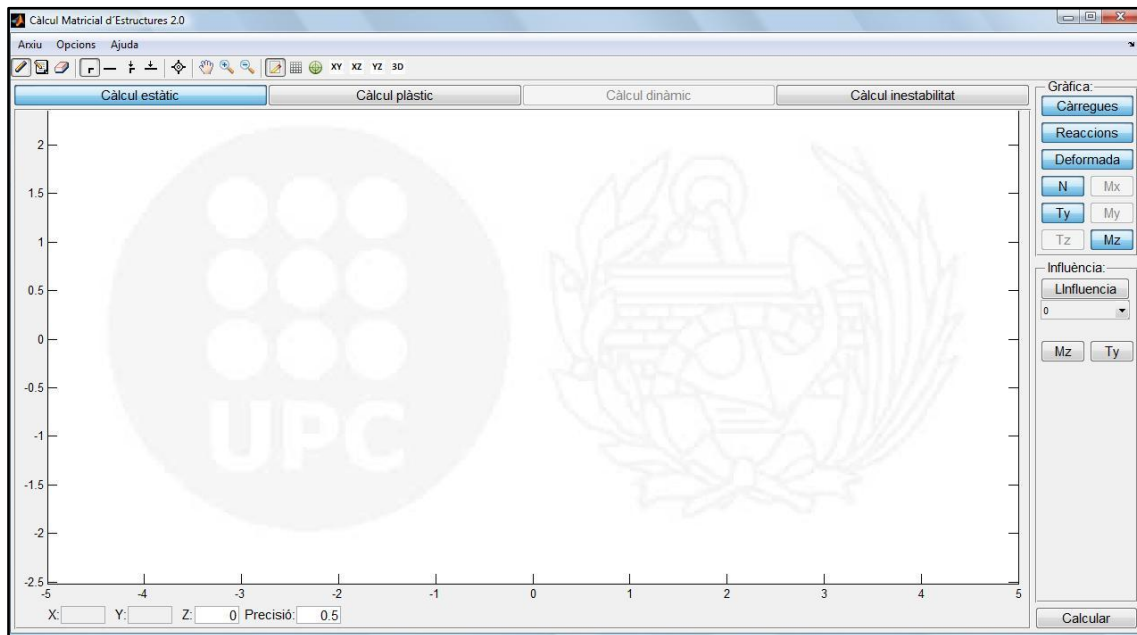


Figura 1. Ventana de inicio al ejecutar cme2

En primer lugar, abrimos el programa cme2 y se creará la ventana del programa. La siguiente imagen es la vista de la ventana del programa antes de ejecutar las primeras funciones del módulo de influencia (Figura 2).

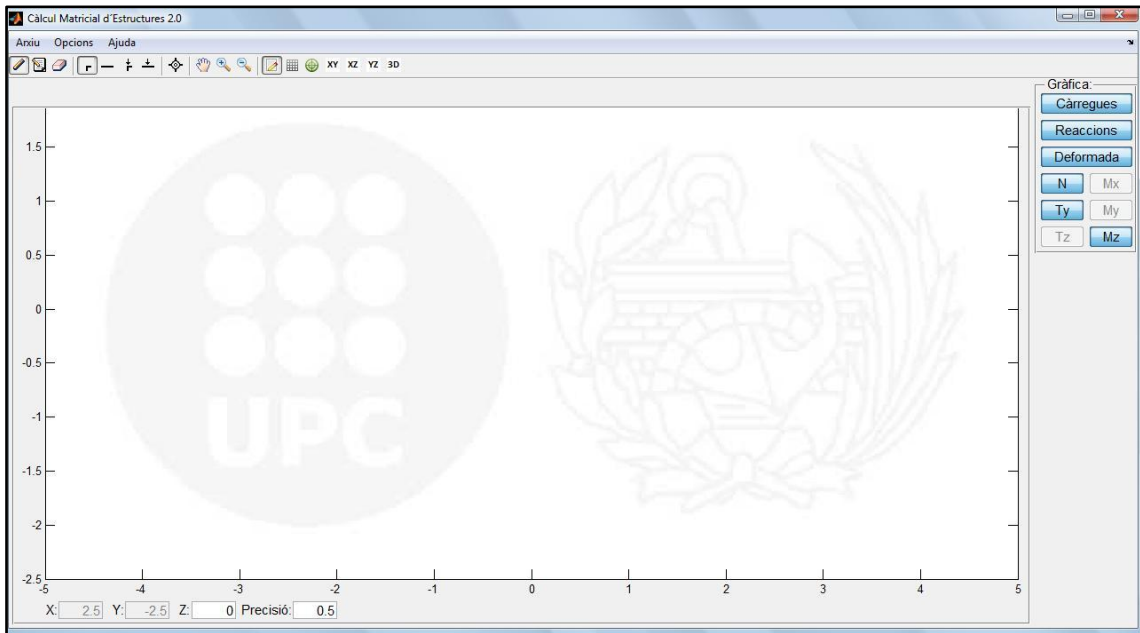


Figura 2. Pantalla de inicio del programa

Lo primero que se creará será el panel de influencia que se encuentra debajo del panel de los botones de las leyes de esfuerzo para el cálculo estático. Para ello será llamada la función `panel_botons_influencia` que creará el siguiente recuadro con sus respectivos botones (Figura 3).

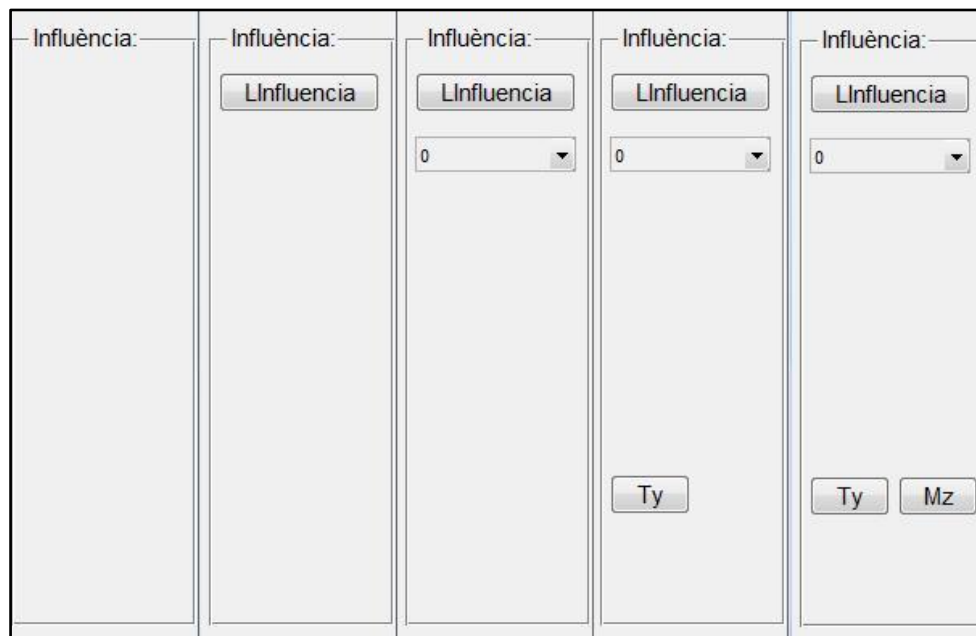


Figura 3. Creación del panel de influencia paso a paso

Tal y como se ve en la imagen, primero se crea el recuadro donde irán los botones, llamado `panelInf`, en segundo lugar se crea el botón de influencia, `LInfluencia`, en tercer lugar el pop-up colocado debajo del botón de influencia, en donde irán los resultados de las líneas de influencia de los apoyos y, por último, los botones de las líneas de influencia del cortante en el

eje Y y del momento flector en el eje Z que están situados en la parte inferior del panel, denominados Ty y Mz, respectivamente. Con la función panell_botons_influencia concluiría la inicialización del programa.

3. Activación cálculo de influencia

Una vez ya iniciado el programa, se espera a que el usuario interactúe con éste. En este trabajo, pondremos un sencillo ejemplo de una viga de dos vanos de 5 metros y 10 metros, respectivamente, con tres apoyos simples para ver el funcionamiento del módulo (Figura 4).

Cuando ya tengamos dibujada nuestra estructura debemos elegir la sección que queremos observar a la hora de calcular las líneas de influencia. Para ello se debe colocar una carga puntual en la sección elegida. En nuestro ejemplo, pondremos la carga en la mitad del segundo vano, a una distancia de 5 metros del segundo apoyo. Como queremos calcular las líneas de influencia, esta carga debería ser unitaria pero en caso de que se ponga una carga puntual no unitaria el módulo la hará unitaria dado que queremos calcular las líneas de influencia.

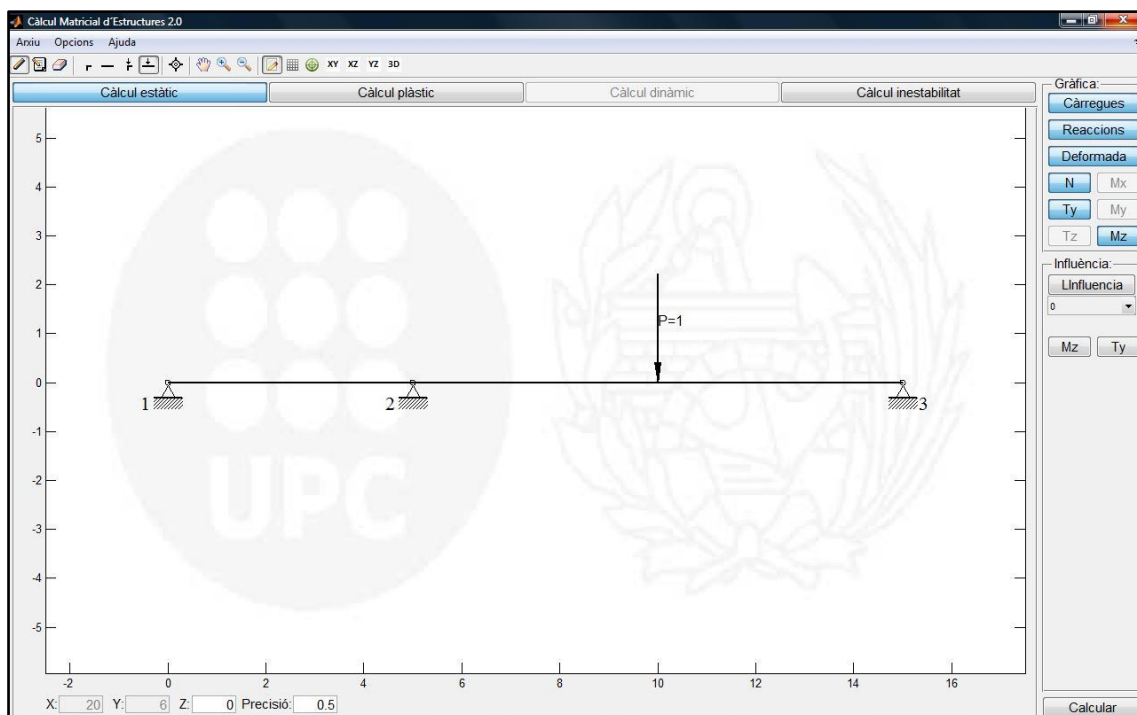


Figura 4. Viga ejemplo con sección a estudiar en la mitad del segundo vano

Ya dibujada nuestra estructura debemos indicar que queremos calcular las líneas de influencia. Para ello tenemos que cumplir dos requisitos. El primero es estar en la ventana de

cálculo estático y, el segundo, seleccionar el botón L. Influència. Después de esto sólo tendremos que apretar el botón Calcular y el programa ejecutará el módulo.

Al apretar el botón Calcular se ejecuta la función `tipus_calcul`. Como estamos en la pestaña de cálculo estático y el botón L. Influència está seleccionado llamará a la función `calcular_influencia`. En el caso de que este botón no estuviera seleccionado, simplemente calcularía y dibujaría las leyes de esfuerzos.

3.1. Preparación de datos y creación de nuevas variables

La función `calcular_influencia` comienza ejecutando `prepara_dades_calcul_influencia` para extraer la información necesaria que se usará a lo largo del módulo y ponerla en el formato de Matlab. Además, se define la variable `barresinicial`, que será usada al final de módulo, `Sec` que es la distancia que hay entre la carga y el primer nudo de la barra donde está situada la sección de estudio y pone a cero las variables: `longT`, que es la suma de todas las barras, `Tmax`, `Mmax` y `Rmax`. Estas tres últimas serán utilizadas para calcular la escala del dibujo. La variable `Sec` puede ir desde cero hasta la longitud de la barra estudiada. En este caso, `Sec` tiene un valor de 5.

También, se define la variable `diagramesinfluencia` en donde será guardada la información para representar las gráficas. En el caso de que ya se haya hecho algún otro cálculo, como calcular las leyes de esfuerzos y ahora se quieran calcular las líneas de influencia volviendo a pulsar sobre el botón calcular, se borrarán todos los diagramas dibujados con la función `esborra_diagrames_influencia_i_reaccions`.

3.2. Cálculo y creación de las líneas de influencia

A partir de este momento, comenzará un loop donde se calcularán, para cada barra, las líneas de influencia respecto la sección seleccionada. Dentro del loop se calcula la longitud de la barra con la variable `long` y pone a cero todas las cargas. Esto se hace por si erróneamente se han introducido alguna carga que no es puntual o algún momento para que, en ese caso, no haya ningún problema o también, por si se ha hecho el cálculo estático y se quiere aprovechar la carga puntual dibujada para calcular las líneas de influencia respecto esa sección.

Después, se definen unos puntos equidistantes por los cuales pasará una carga unitaria puntual (Figura 5) y se calcularán las leyes de esfuerzos respecto la sección inicial dada (carga puntual roja). Este será el valor que se guardará para más tarde dibujarlo en la ventana del Matlab aplicando la definición exacta de línea de influencia.

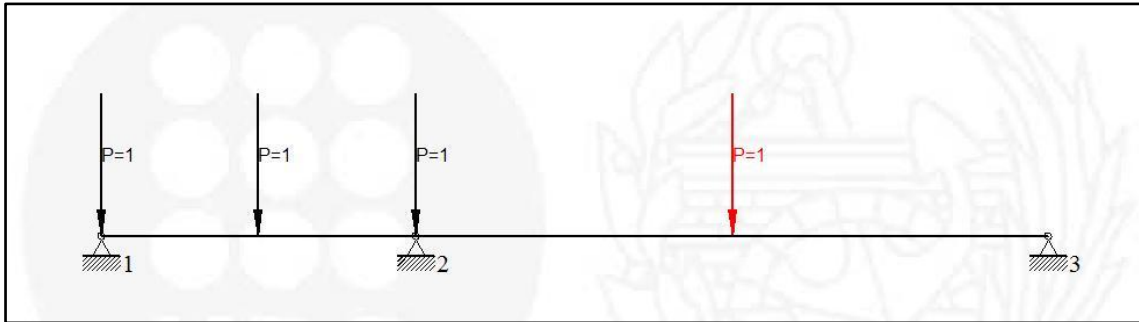


Figura 5. Carga puntual moviéndose por los puntos equidistantes respecto una sección

3.2.1. Cálculo del momento flector, cortante y reacciones

Para cada punto equidistante se seguirá el mismo procedimiento, que se explica a continuación. Una carga unitaria es colocada sobre un punto i -ésimo y cuando se ejecuta la función `cme2_calcul_estatic` se calculan las reacciones en los apoyos, que luego servirán para dibujar las líneas de influencia en éstos. Después, la función `calcul_tallant_moment_reacions` calcula el polinomio de las leyes de esfuerzos (cortante y momento flector) de la barra donde se encuentra la sección. En este ejemplo, estamos dibujando las líneas de influencia de la primera barra, pero calculando las leyes de la segunda barra que es donde se encuentra nuestra sección. Así que, una vez se tienen los polinomios, la función calcula el valor del cortante y del momento flector en la sección seleccionada (carga roja) que ha sido generada al aplicar una carga unitaria y puntual en ese punto i -ésimo. Esta información se guarda en dos vectores que se van actualizando para cada punto.

Una vez calculados el cortante y el momento flector, también se calculan las reacciones de los apoyos generadas por una carga puntual situada en el punto i -ésimo guardando esta información para ser utilizada más adelante. Cuando ya hemos obtenido toda la información del valor del cortante, momento flector y reacciones en los apoyos de dicho punto, se elimina la carga unitaria situada en el punto i -ésimo y se reinicia el loop para situar la misma carga unitaria en el punto $i+1$ y volver a calcular las leyes de esfuerzo y reacciones respecto la sección de estudio. El loop va trabajando punto por punto hasta recorrer toda la barra 1, desde el nodo 1 hasta el nodo 2 como se ha visto en la figura anterior.

En nuestro caso los valores obtenidos para diez puntos equidistantes son:

Momento flector

0	-0,0457	-0,0880	-0,1235	-0,1486	-0,1600	-0,1543	-0,1280	-0,0777	0
---	---------	---------	---------	---------	---------	---------	---------	---------	---

Cortante

0	-0,0091	-0,0176	-0,0247	-0,0297	-0,0320	-0,0309	-0,0256	-0,0155	0
---	---------	---------	---------	---------	---------	---------	---------	---------	---

Reacciones

1,0000	0,8706	0,7426	0,6173	0,4961	0,3804	0,2716	0,1710	0,0800	0,0000
0,0000	0,1385	0,2750	0,4074	0,5336	0,6516	0,7593	0,8546	0,9355	1,0000
0,0000	-0,0091	-0,0176	-0,0247	-0,0297	-0,0320	-0,0309	-0,0256	-0,0155	0,0000

Posición puntos

0,0000	0,5556	1,1111	1,6667	2,2222	2,7778	3,3333	3,8889	4,4444	5,0000
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Finalmente, el último cálculo realizado es la posición del cortante en apoyos y sección. En nuestro caso, todavía estamos trabajando la primera barra y, por lo tanto la posición del valor del cortante coincide con la posición de los puntos equidistantes, pero hay que tener en cuenta que esto no sucederá cuando estemos trabajando la segunda barra, por ese motivo se va a explicar cómo el programa forma este vector.

Suponiendo que ya hemos calculado toda la información de la primera barra, el loop pasaría a la segunda barra y procedería de la misma forma explicada anteriormente hasta llegar a los puntos que determinan la posición del cortante. En el caso de que nos encontremos en la barra de la sección elegida hay dos posibles situaciones:

- a) La carga está aplicada en algún punto intermedio de la barra, por lo tanto, sabemos que en la sección estudiada habrá un cambio de signo del cortante, es decir, en un mismo punto el cortante tiene dos valores de signo contrario. Estos dos valores han sido calculados, por ese motivo se ha de cambiar el vector de puntos equidistantes e incluir un punto más para que a la hora de dibujar la gráfica se dibuje correctamente.
- b) La carga está aplicada en los apoyos con lo que las imágenes de los cortantes coinciden con los puntos equidistantes.

Como nosotros estamos todavía en la primera barra, en este caso, no habrán dos gráficas en una misma barra con un cambio de signo del cortante porque empezará desde el apoyo 1 hasta el apoyo 2, pero en la segunda barra sí, habrán dos cortantes en un mismo punto dividiendo la gráfica en dos gráficas, una desde el primer apoyo hasta la sección seleccionada con un signo y otra desde la sección seleccionada hasta el siguiente apoyo con signo contrario. Esto se tendrá en cuenta para que a la hora de dibujar la gráfica aparezca un cortante desde el apoyo 2 hasta el punto donde está la sección y desde ese mismo punto otro cortante hasta el apoyo 3. Sabiendo que, la suma de ambos cortantes en la sección de estudio ha de ser igual a la unidad.

3.2.2. Función principal: dibujo de los diagramas de influencia

Una vez hemos obtenido todos los valores necesarios, el programa pasa a ejecutar la función `dibuix_diagrames_reaccions`. Esta función empieza creando la variable `diag` que genera todos los diagramas de influencia, vacíos de momento. Después se empieza el loop de las líneas de influencia de las reacciones. En este caso, habrá tres líneas de influencia por haber tres apoyos.

Para cada apoyo se llama a la función `dibudiag_reaccions_moment` (`txt,P,coordP, long,detall,signe`) donde `txt` es el texto que saldrá representado en el diagrama, dado que se están calculando las líneas de influencia de las reacciones, `txt` estará representado por `R`; `P` es la información que hemos calculado anteriormente para ese apoyo, es decir, para cada punto equidistante hemos calculado la reacción en el apoyo 1 cuando una carga puntual unitaria se desplaza a lo largo de éstos, por tanto `P` son esos valores calculados; `coordP` son los puntos equidistantes; `long` es la longitud de la barra; `detall` es un valor para calcular los mínimos y los máximos y, finalmente, `signe` es el criterio de signos que queremos usar para dibujar nuestro diagrama en la parte de arriba o en la parte de abajo según signo positivo o negativo. Nosotros, cogeremos el mismo criterio de signos que el usado en el cálculo estático (Figura 6).

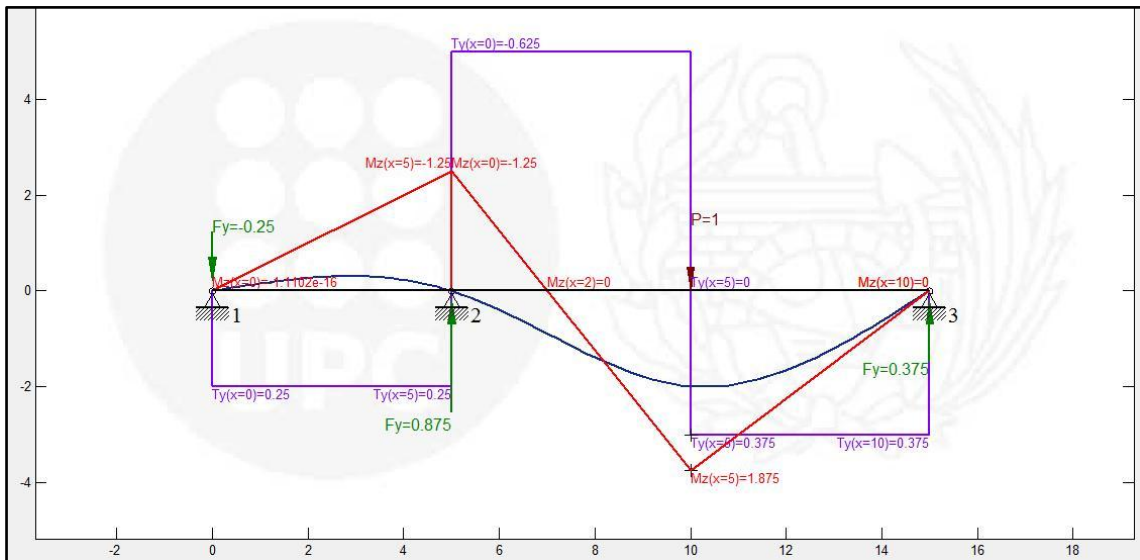


Figura 6. Criterio de signos para momento flector y cortante

Se definen las variables `hg` que es donde se guardará la gráfica, las variables `miny` y `maxy` para calcular los valores máximos y mínimos y `xd` y `yd` que serán los puntos que uniremos para dibujar la gráfica. A partir de aquí la función utiliza la información calculada.

Tenemos evaluados unos puntos equidistantes con sus respectivos valores y además se añadirán dos puntos más, uno inicial y otro final que tendrán valor cero para que, una vez unidos todos los puntos la gráfica empiece en el primer nodo de la barra y termine en el último (Figura 7).

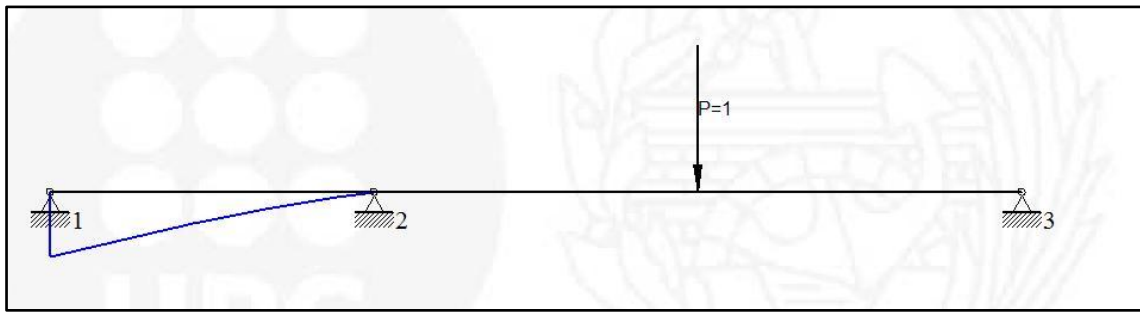


Figura 7. Línea de influencia del apoyo 1 de la primera barra

Una vez dibujada la gráfica se pasa a calcular el texto. Para ello se buscan los mínimos y máximos absolutos y los valores iniciales y finales. En el apoyo 1 si ponemos una carga puntual unitaria en éste, el valor de la reacción es uno. A medida que la carga unitaria se va alejando del apoyo uno, el valor de la reacción de éste disminuye hasta que la carga unitaria está sobre el apoyo 2 y, por lo tanto, la reacción del apoyo 1 será nula. Quedando de la siguiente forma (Figura 8).

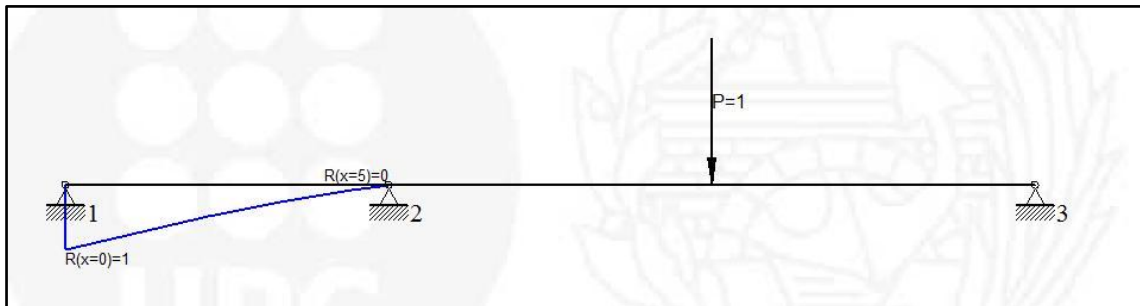


Figura 8. Texto de la línea de influencia

Y de esta forma ya se habrán dibujado las líneas de influencia que suceden en el primer nudo de la primera barra cuando pasa una carga unitaria uniforme desde el apoyo 1 hasta el apoyo 2. De la misma forma se procede para el apoyo 2 de la primera barra y el apoyo 3 de la segunda y se dibuja el diagrama restante quedando definidos todos los diagramas de las líneas de influencia de las reacciones en la primera barra (Figura 9).

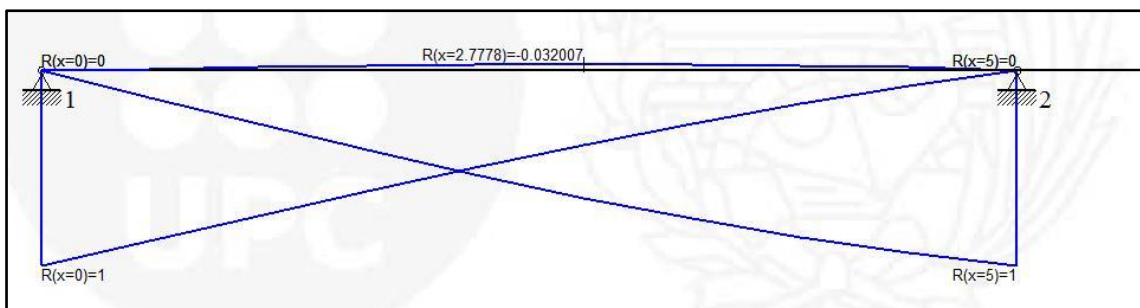


Figura 9. Líneas de influencia de las reacciones de la primera barra

Una vez se han dibujado las reacciones, se pasa a calcular las líneas de influencia del cortante y el momento flector de forma similar con las funciones `dibudiag_tallant('Ty', Ty, PrTy, long, 1, 1)` y

dibudiag_reaccions_moment('Mz',Mz,Pr,long,1,1), respectivamente (Figura 10). También se calculan los máximos y los mínimos para calcular la escala de la gráfica.

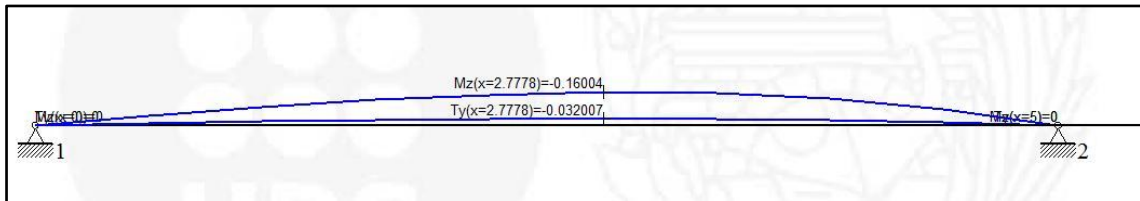


Figura 10. Líneas de influencia del cortante y del momento flector de la primera barra

Cuando ya se han dibujado todas las líneas de influencia, se indica el número de la barra en las cuales han sido calculadas mediante `diagramasinfluencia(ii).num=ii` con `ii` el número de la barra y también se calculan las variables `Tmax`, `Mmax` y `Rmax` a partir de los máximos y mínimos de las gráficas.

Todo lo mencionado anteriormente se repetirá para la segunda barra obteniendo todas las líneas de influencia (Figura 11).

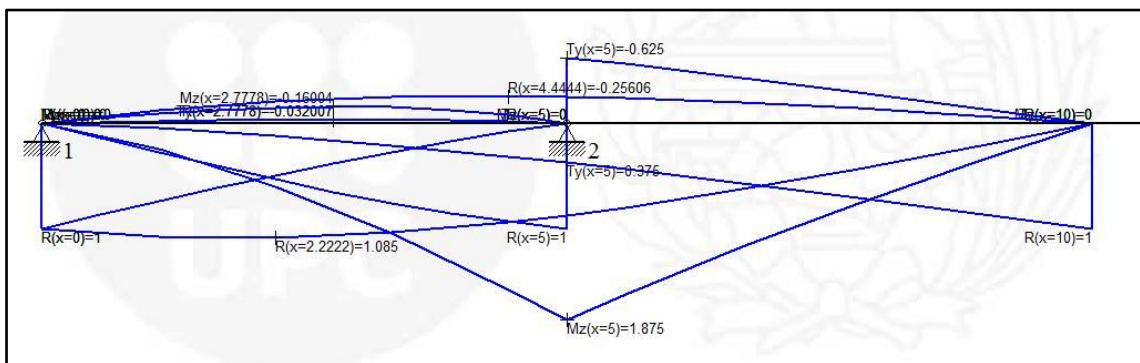


Figura 11. Líneas de influencia de las reacciones, cortante y momento flector

Como podemos observar todas las líneas de influencia están mal colocadas y sin escala, por ese motivo a continuación las recolocaremos, les daremos color y las escalaremos correctamente. Para la escala definimos tres variables `sT`, `sM` y `sR` que nos servirán para adecuar la escala de cada línea de influencia de forma individual dado que, en algunos casos, pueden generarse líneas de influencia de momentos muy pequeñas o muy grandes comparadas con las reacciones y si pusiéramos todo en la misma escala, algunas líneas quedarían cortadas por el borde de la ventana del programa y no quedaría estético.

3.2.3. Colocación, escala y coloreado de los diagramas de influencia

Se definen las variables `pos`, `vector_barra`, `mT`, `mM` y `mR`. La variable `pos` coloca el diagrama en la posición del nudo inferior, dado que por defecto, todos los gráficos se dibujan en las coordenadas $(X,Y)=(0,0)$. La variable `vector_barra` indica la dirección de la barra. Al tratar

Ejemplo recolocación de las líneas de influencia

Diagrama de un sistema de vigas con un apoyo fijo en el extremo izquierdo y cuatro apoyos móviles en los puntos 2, 3, 4 y 5. Una carga puntual $P=1$ actúa hacia abajo en el punto 2. El eje horizontal está etiquetado con valores de -4 a 1.

17

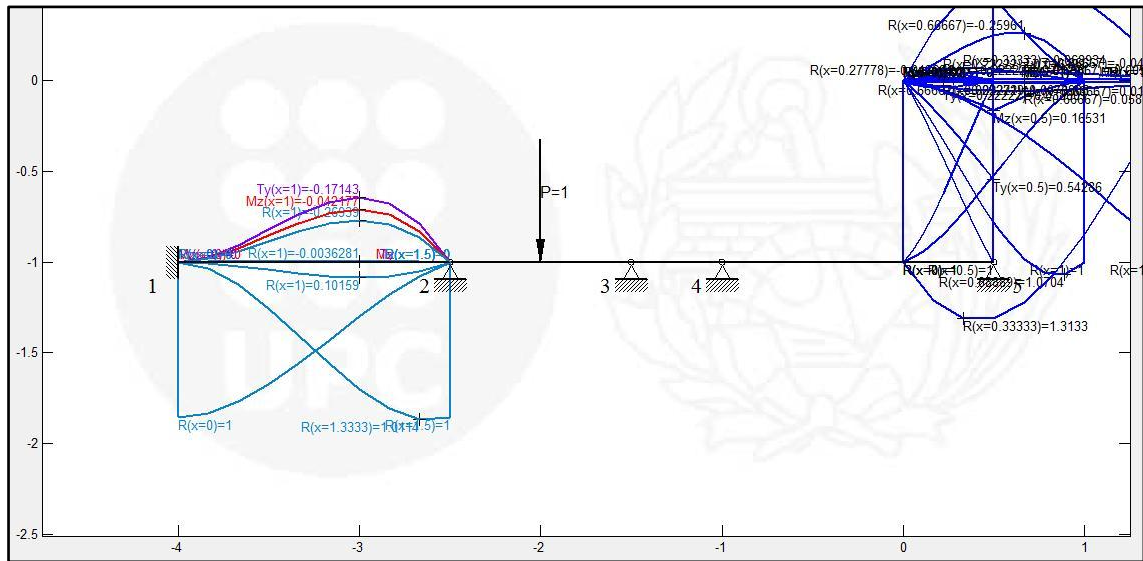


Figura 14. Recolocación de las líneas de influencia de la primera barra del ejemplo anterior

Y finalmente se da color al diagrama con la función colordiagrama_influencia. En donde se ha optado por usar para las líneas de influencia del cortante en el eje Y y del momento flector en el eje Z los mismos colores que las leyes de esfuerzos obtenidas en el cálculo estático. El color violeta para el cortante y el color rojo para el momento flector. Para las líneas de influencia de las reacciones se ha usado el color azul (Figura 15).

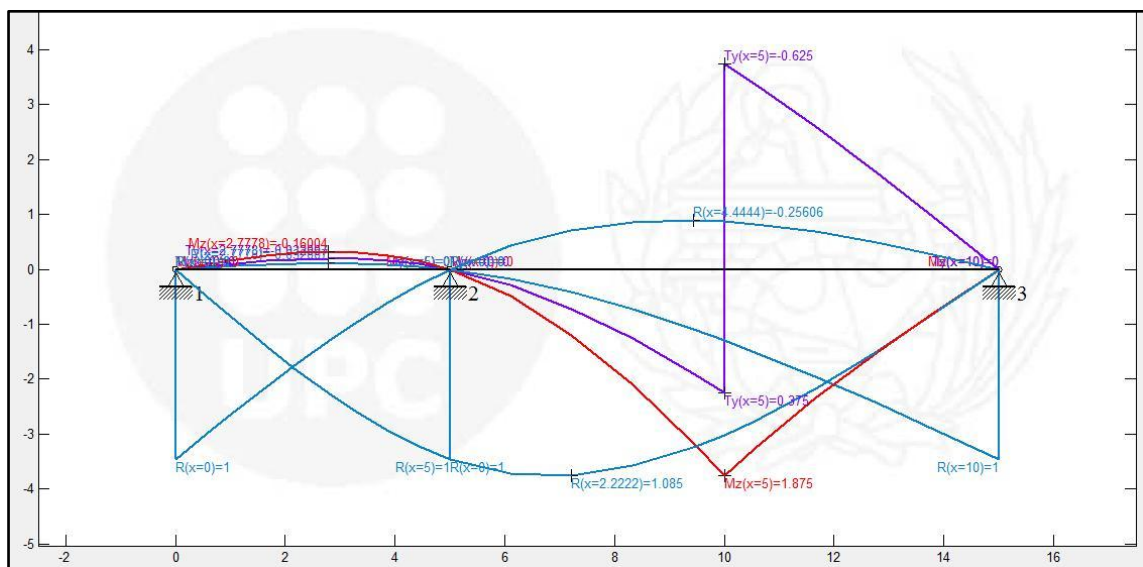


Figura 15. Representación final de las líneas de influencia

3.2.4. Activación botones y visibilidad diagramas

Después de haber dibujado las líneas de influencia se han de configurar los botones creados en la inicialización del programa para mostrar u ocultarlas. Para relacionar los diagramas con los botones se escribe un comando que actualiza el pop-up de las líneas de influencia de las reacciones (Figura 16) y se llama a la función `visibilitat_diagramesInf` para poder mostrar u ocultar las líneas de influencia del cortante y del momento flector. También se activan los botones `Ty` y `Mz` para que aparezcan todas las líneas de influencia (Figura 17).



Figura 16. Actualización del pop-up de las líneas de influencia de las reacciones

3.2.5. Presentación de los resultados

Para finalizar se redibuja la carga puntual inicialmente dibujada en la sección de estudio, y que durante la ejecución del programa se ha eliminado, para indicar que las líneas de influencia del cortante y del momento flector han sido calculadas respecto esa sección. Esto se hace llamando a la función `redibuixar_carregues_barres`. La función `edicio` desactiva los botones de la barra de herramientas para la edición y activa la muestra de los resultados mostrando todas las líneas de influencia (Figura 17).

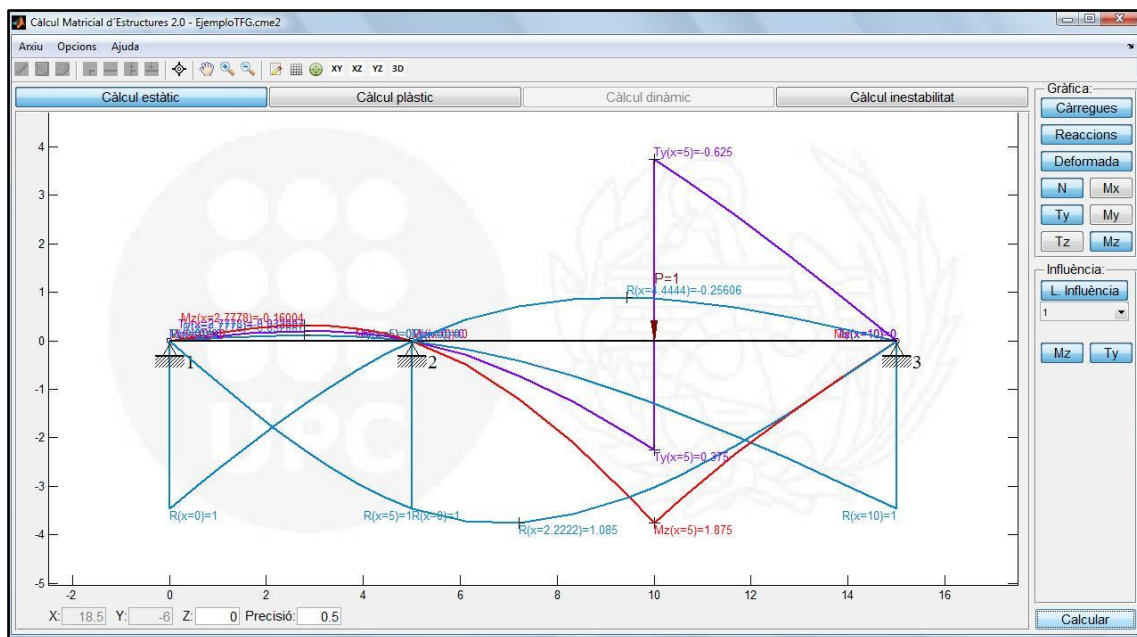


Figura 17. Ventana final después de ejecutar el módulo de influencia

CAPÍTULO 2. Explicación del código en MATLAB

```

function cme2()

%% CREACIÓ I ACCIONS SOBRE EL PANELL QUE CONTÉ ELS BOTONS QUE CONTROLLEN EL CàLCUL DE LES LÍNIES D'INFLUÈNCIA
function
[panelInf,botoCalculInfluencia,popupMenuInf,bototallantYInf,botomomentZInf]=panell_botons_influencia(contbarra)
function popupcallbackinf(hObject,~,~)
function autopopupcallbackinf(val)
function diagTyInf(~,~,~)
function diagMzInf(~,~,~)
function visibilitat_diagramesInf

%% CREACIÓ I ACCIONS SOBRE LA FINESTRA PRINCIPAL, REDIMENSIONAMENT, CONTROL DEL RATOLÍ I TECLAT, ETC:
function figcme2_resize(~,~,~)

%% CREACIÓ I ACCIONS SOBRE EL PANELL QUE CONTÉ ELS BOTONS DELS DIFERENTS TIPUS DE CàLCUL.
function calcul_estatic(hObject,~,~)
function calcul_dinamic(hObject,~,~)
function calcul_plastic(hObject,~,~)
function calcul_inestabilitat(hObject,~,~)

%% CREACIÓ I ACCIONS QUE S'EXECUTEN AMB EL BOTÓ CALCULAR.
function tipus_calcul(~,~,~)
function calcularinf

%% FUNCIONS MÒDUL D'INFLUÈNCIA
function [hg,miny,maxy]=dibudiag2(txt,P,coordP,P0,sep,long,detall,signe)
function [hg,miny,maxy]=dibudiag1(txt,P,coordP,P0,sep,long,detall,signe)
function txtdiag(x,y,txt,long,hg,signe)
function colordiagramareaccions(diag)
function [N,Ty,Tz,Mx,My,Mz]=calcul_tallant_moment_reaccions(seccion,ii)
function [long,esforços,moviments,rep,proj,punt,mom,Iz,Iy]=dades_diagramainfluencia(n)
end

```

Código 0. Función cme2

Este capítulo se centra en el desarrollo del código del programa, explicando sus funciones variables y su funcionamiento. Para la comprensión de este capítulo se requieren ciertas nociones de Matlab de MathWorks, Inc. aunque se ha intentado explicar de una forma clara y sencilla para que cualquier usuario sea capaz de entenderlo.

Como se decía anteriormente en la introducción, el código del programa se ha desarrollado mediante el software Matlab. Este trabajo incorpora al programa de análisis de estructuras con interfaz gráfica CME 2.0. un módulo para calcular líneas de influencia. El programa incluía cuatro módulos; el de análisis estático, análisis plástico, análisis de inestabilidad y análisis dinámico. Con este nuevo módulo de líneas de influencia, queda un programa más completo para ayudar a la docencia.

De este modo, el código del módulo de influencia se ha desarrollado en un código ya existente. Se ha intentado crear un código independiente a los otros módulos, exceptuando algunas funciones y variables de la interfaz gráfica relativas a la creación de la estructura e introducción de datos, que se han dispuesto en el script principal `cme2.m`.

Se mostrará todo el código usado en el módulo del cálculo de las líneas de influencia y se explicarán todas las funciones detalladamente haciendo hincapié en aquellas que no hayan sido explicadas en el capítulo anterior.

En el Código 0 podemos ver la función `cme2`. Con lo que siguiendo las funciones del programa dividiremos este capítulo en cuatro puntos:

1. Creación y acción del panel y botones de influencia. Definición del entorno gráfico
2. Redimensionamiento de la ventana del programa
3. Cambios en la activación de los botones de cálculo
4. Acción que se ejecuta con el botón calcular y construcción del diagrama de influencia

1. Creación y acción del panel y botones de influencia. Definición del entorno gráfico

```
function
[panelInf,botoCalculInfluencia,popupMenuInf,bototallantYInf,botomomentZInf]=panell_botons_influencia(contbarra)
    contbarra=contbarra+1;
    numerobarres=zeros(contbarra,1);
    panelInf=uipanel(figcme2,'Units','pixels','Position',[tm(1)-
115,35,115,375],'FontName','default','BorderType','etchedin','BorderWidth',1,'FontUnits','pixels','FontSize',14,'Ti
tle','Influència:'); % Creació del panell d'influencia.
    botoCalculInfluencia=uicontrol(panelInf,'Units','pixels','Position',[5,tm(2)-
300,100,25],'Style','togglebutton','FontName','default','FontUnits','pixels','FontSize',14,'String','LInfluencia','
Value',0,'Callback',@calcul_estatic,'KeyPressFcn',@tecla); % Creacio botó càlcul línies influència.
    popupMenuInf = uicontrol(panelInf,'Style','popupmenu','String',numerobarres,'Position',[5,tm(2)-
350,100,35],'callback', @popupcallbackinf); % Creació del popupmenu amb les línies d'influencia de les reaccions.
    bototallantYInf=uicontrol(panelInf,'Units','pixels','Position',[5,tm(2)-
550,50,25],'Style','togglebutton','FontName','default','FontUnits','pixels','FontSize',14,'String','Ty','Value',0,'
Callback',@diagTyInf,'KeyPressFcn',@tecla); % Creacio botó vista de diagrames del tallant en y.
    botomomentZInf=uicontrol(panelInf,'Units','pixels','Position',[60,tm(2)-
550,50,25],'Style','togglebutton','FontName','default','FontUnits','pixels','FontSize',14,'String','Mz','Value',0,'
Callback',@diagMzInf,'KeyPressFcn',@tecla); % Creacio botó vista de diagrames del moment en z.
end
```

Código 1. Función panell_botons_influencia

```
function popupcallbackinf(hObject,~,~)
%Es la funció que es crida quan es selecciona alguna opció del popup de les línies d'influència de les reaccions.

val = get(hObject,'Value'); % Agafa el valor del popup que correspon al número de reacció que es vol estudiar.
autopopupcallbackinf(val); % Aquesta funció permet actualitzar el popup i efectuar tots els càlculs necessaris per
mostrar els resultats.
end
```

Código 1.1.Función popupcallbackinf


```

function autopopupcallbackinf(val)
    % Permet cridar a la mostra de resultats de les reaccions automàticament.

    if get(botoCalculInfluencia, 'Value')
        vis='on';
    else
        vis='off';
    end
    if grafica==grafica_est
        for i=1:numel(diagramesinfluencia)
            set(diagramesinfluencia(i).hgR(val), 'Visible', vis)
        end
    else
        set(diag_presxy.hgMz, 'Visible', vis)
    end
end
end

```

Código 1.1.1. Función autopopupcallbackinf

```

function diagTyInf(~,~,~)
    % Oculta o mostra el diagrama d'influència del tallant en y a la representació de resultats.
    if get(bototallantYInf, 'Value')
        vis='on';
    else
        vis='off';
    end
    if grafica==grafica_est
        for i=1:numel(diagramesinfluencia)
            set(diagramesinfluencia(i).hgTy, 'Visible', vis)
        end
    else
        set(diag_presxy.hgTy, 'Visible', vis)
    end
end
end

```

Código 1.2.Función diagTyInf

```

function diagMzInf(~,~,~)
    % Oculta o mostra el diagrama d'influencia de moments en z a la representació de resultats.

    if get(botomomentZInf, 'Value')
        vis='on';
    else
        vis='off';
    end
    if grafica==grafica_est
        for i=1:numel(diagramesinfluencia)
            set(diagramesinfluencia(i).hgMz, 'Visible', vis)
        end
    else
        set(diag_presxy.hgMz, 'Visible', vis)
    end
end
end

```

Código 1.3. Función diagMzInf

```

function visibilitat_diagramesInf
    % Oculta o mostra el conjunt de diagrames d'influencia.

    diagTyInf
    diagMzInf
end

```

Código 1.4. Función visibilitat_diagramesInf

1. Función panell_botons_influencia

Esta función crea el panel de influencia y los botones que muestran las líneas de influencia. En total son tres botones y un pop-up para las líneas de influencia de las reacciones.

En primer lugar se crea el recuadro panellInf que está situado en la ventana figcme2, después se crea el popupMenuInf que sirve para representar las líneas de influencia de las reacciones en los apoyos y por último, se crean los botones de las líneas de influencia del momento flector en el eje Z y del cortante en el eje Y.

En el recuadro popupMenuInf aparecerán tantos números como apoyos tengamos. En el caso de tener dos vigas con cuatro apoyos aparecerán los números del 1 al 4, una vez hayamos dado al botón calcular. Cuando se hace click en alguno de estos números se llamará a la función popupcallbackinf que mostrará u ocultará el diagrama de influencia de dicho número.

No se mostrarán las líneas de influencia de los nodos libres ni rótulas ya que no son apoyos.

El pop-up de las líneas de influencia de las reacciones está situado bajo el botón de Linfluencia y los otros dos botones por debajo del pop-up. Por defecto Linfluencia está activado.

1.1. Función popupcallbackinf

Esta es la función que se llama cuando se selecciona alguna opción del pop-up de las líneas de influencia de las reacciones.

1.1.1. Función autopopupcallbackinf

Cuando se hace click en el pop-up de las líneas de influencia de las reacciones esta función controla que el diagrama sea mostrado o no. Si el botón de las líneas de influencia, Linfluencia, está seleccionado al hacer click en cualquier apoyo aparecerán las líneas de las reacciones de cada apoyo para la sección seleccionada. En el caso de que el diagrama ya haya aparecido, si el botón Linfluencia sigue activado y se hace click en algún número del pop-up correspondiente con algún apoyo, no pasará nada. Si se desactiva el botón Linfluencia y se vuelve al pop-up seleccionando cualquier apoyo, esta línea de influencia de ese apoyo será ocultada. Es decir, esta función hará que se muestren u oculten los resultados obtenidos gráficamente dependiendo de si el botón Linfluencia está seleccionado o no. Veamos un ejemplo:

Ejemplo activación/desactivación líneas de influencia

Cuando damos al botón calcular se nos calcularán todas las líneas de influencia. En este ejemplo hemos desactivado las líneas del momento flector y cortante para centrarnos en las líneas de las reacciones (Figura 18). Vemos que, por defecto, se dibujan todas las líneas de las reacciones. Si

nosotros sólo queremos ver una de ellas, por ejemplo la línea de influencia del apoyo 3, tenemos que desactivar el botón LInfluencia y apretar el número 2 del pop-up para ocultar el diagrama (Figura 19).

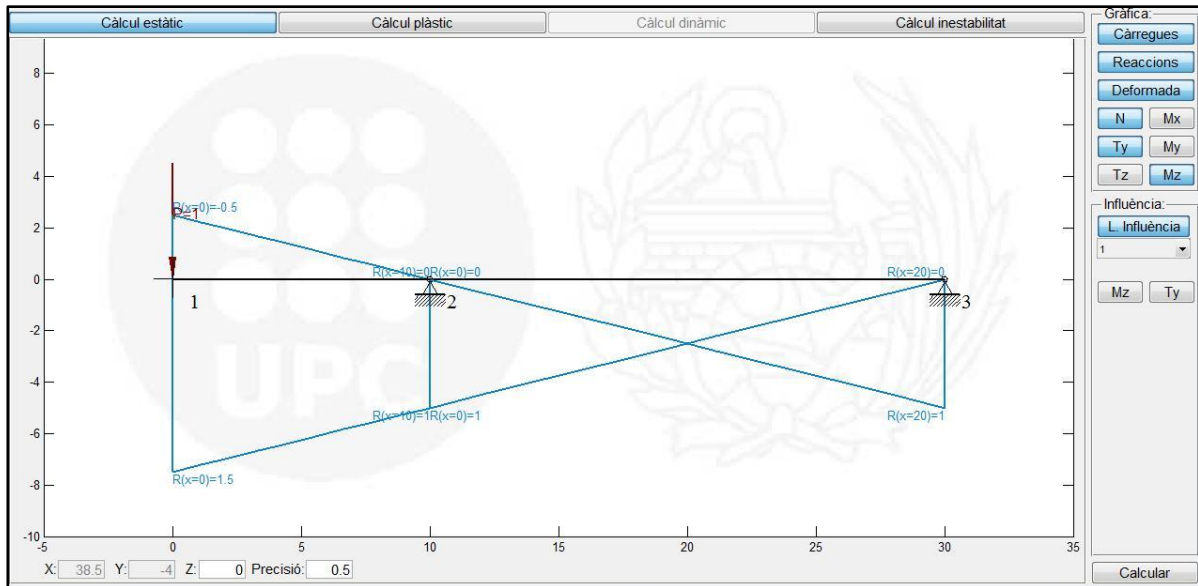


Figura 18. Ejemplo pop-up líneas de las reacciones

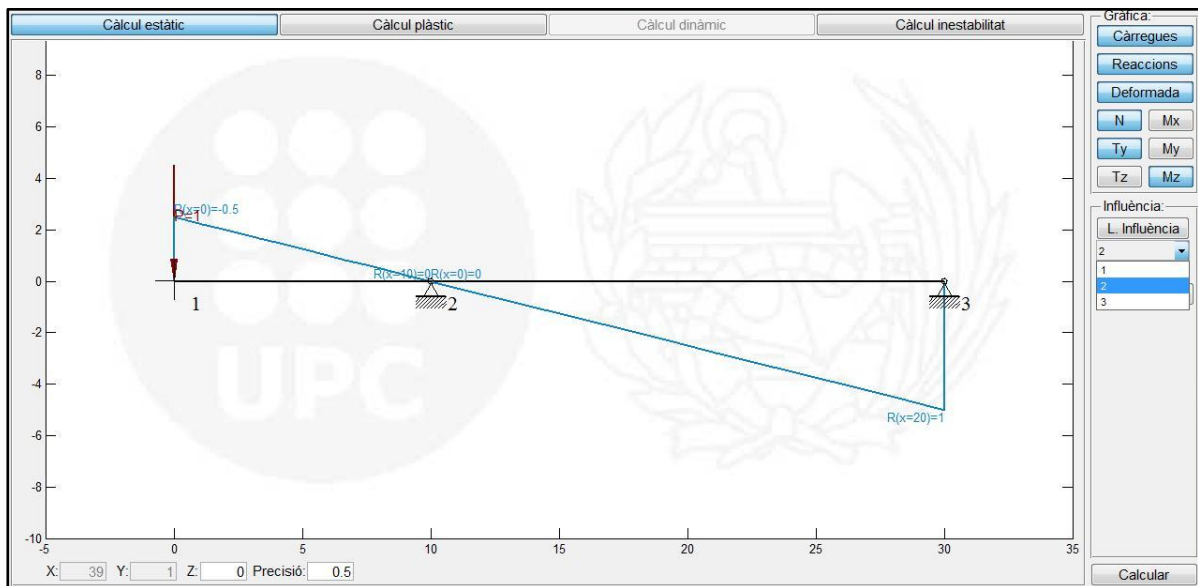


Figura 19. Ejemplo desactivación línea influencia del apoyo 2

1.2. Función diagTyInf

Esta función muestra u oculta los resultados obtenidos por el programa y funciona de forma distinta a las líneas de influencia de las reacciones.

Una vez se ha apretado el botón calcular, por defecto, todos los diagramas serán mostrados. Si nosotros apretamos el botón de las líneas de influencia del cortante en el eje Y, la gráfica desaparecerá. Si se vuelve a pulsar este botón, mostrará otra vez la gráfica.

A diferencia del pop-up de las líneas de influencia de las reacciones, este botón no depende del botón LInfluencia.

1.3. Función *diagMzInf*

Tiene la misma utilidad que la anterior función. La diferencia es que esta función está enlazada con el botón de las líneas de influencia del momento flector en el eje Z y al pulsar dicho botón se mostrará u ocultará la gráfica de las líneas de influencia del momento flector para la sección escogida.

1.4. Función *visibilitat_diagramesInf*

Hace que los diagramas de las líneas de influencia, después de que sean calculados, sean mostrados o no. Por defecto, después de apretar el botón Calcular aparecerán todas las líneas de influencia.

2. Redimensionamiento

```
function figcme2_resize(~,~,~)
% Recoloca els panells quan es redimensiona la finestra principal del
% programa.

% Panell de les línies d'influència.
set(panelInf, 'Position', [tm(1)-115,35,115,tm(2)-233]);

% Pop-up del panell de les línies d'influència de les reaccions.
set(popupMenuInf, 'Position', [5,tm(2)-480,100,200]);

% Botó del càlcul de les línies d'influència.
set(botoCalculInfluencia, 'Position', [5,tm(2)-280,100,25]);

% Botó de les línies d'influència del moment flector en Z.
set(botomomentZInf, 'Position', [5,tm(2)-350,50,25]);

% Boto de les línies d'influència del tallant en Y.
set(bototallantYInf, 'Position', [60,tm(2)-350,50,25]);
end
```

Código 2. Función *figcme2_resize*

Esta función está diseñada para redimensionar los paneles y botones cuando se cambia el tamaño de la pantalla. Al iniciarse el programa siempre se inicia con unos valores determinados y los paneles y botones han sido colocados en una determinada posición como se puede ver en el Código 1. Si nosotros quisiéramos maximizar o minimizar la ventana entonces estos botones quedarían en la posición que han sido puestos, fijados en los píxeles exactos e inmóviles. Para evitar que esto ocurra debemos cambiar la función del Código 2 e incorporar los nuevos paneles y botones a ella para que dependan de las dimensiones de la ventana actual y no de la inicial.

3. Cambios en la activación de los botones de cálculo

```
function calcul_estatic(hObject,~,~) %Opcions que s'activen al apretar
el botó càlcul estàtic
    if get(hObject,'Value')
        %Posa les altres pestanyes en off
        set(botoCalculPlastic,'Value',0);
        set(botoCalculDinamic','Value',0);
        set(botoCalculInestabilitat,'Value',0);
        %Desactiva els panells que no corresponguin al càlcul estàtic
        set(panelPlastic,'Visible','off');
        set(panel3,'Visible','on');
        set(panelInf,'Visible','on');
        set(panelInest,'Visible','off');
        set(panelres2ord,'Visible','off');
    end
```

Código 3.1. Función calcul_estati

```

function calcul_dinamic(hObject,~,~)%Opcions que s'activen al apretar el botó càlcul dinàmic
    if get(hObject,'Value')
        %Posa les altres pestanyes en off
        set(botoCalculEstatic,'Value',0);
        set(botoCalculPlastic,'Value',0);
        set(botoCalculInestabilitat,'Value',0);
        %Desactiva els panells que no corresponguin al càlcul dinàmic
        set(panelPlastic,'Visible','off');
        set(panel3,'Visible','on');
        set(panelInf,'Visible','off');
        set(panelInest,'Visible','off');
        set(panelres2ord,'Visible','off');
    end
end

```

Código 3.2. Función calcul_dinamic

```

function calcul_plastic(hObject,~,~)%Opcions que s'activen al apretar el botó càlcul plàstic
    if get(hObject,'Value')
        %Posa les altres pestanyes en off
        set(botoCalculEstatic,'Value',0);
        set(botoCalculDinamic,'Value',0);
        set(botoCalculInestabilitat,'Value',0);
        set(panel3,'Visible','on');
        set(panelInf,'Visible','off');
        set(panelInest,'Visible','off');
        set(panelres2ord,'Visible','off');
        %Desactiva els panells que no corresponguin al càlcul plastic
        %Activa el panell corresponent al càlcul plàstic
        set(panelPlastic,'Visible','on');
    end
end

```

Código 3.3. Función calcul_dinamic

```

function calcul_inestabilitat(hObject,~,~) %Opcions que s'activen al apretar el botó càlcul inestabilitat
    if get(hObject,'Value')
        %Posa les altres pestanyes en off
        set(botoCalculPlastic,'Value',0);
        set(botoCalculDinamic,'Value',0);
        set(botoCalculEstatic,'Value',0);
        set(panel3,'Visible','off');
        set(panelInf,'Visible','off');
        set(panelInest,'Visible','on');
        try
            if strcmp(get(def2ord,'Visible'),'on');
                set(panelres2ord,'Visible','on');
            end
        catch
            def2ord=line(0,0,0,'Visible','off');
        end
        %Desactiva els panells que no corresponguin al càlcul inestabilitat
        set(panelPlastic,'Visible','off');
    end
end
end

```

Código 3.2. Función calcul_inestabilitat

Se han modificado las funciones anteriores de forma que cuando nos encontremos en la pestaña del cálculo estático aparezca el panel de influencia de forma directa (panelInf, 'Visible', 'on') y que cuando cambiemos a cualquier otro módulo dinámico, plástico o de inestabilidad, desaparezca el panel de influencia (panelInf, 'Visible', 'off').

4. Acción que se ejecuta con el botón calcular y construcción del diagrama de influencia

```
function tipus_calcul(~,~,~)
    % Hem serveix per veure a quina pestanya em trobo.
    if get(botoCalculInfluencia,'Value')
% Si està seleccionada la pestanya de càlcul estàtic i també el botó
d'influència passa a la funció calcularinf
        calcular_influencia;
    else
        if get(botoCalculEstatic,'Value')
% Si esta seleccionada la pestanya de càlcul estàtic passa a la funció calcular
            calcular;
        end
    end
    if get(botoCalculPlastic,'Value')
% Si està seleccionada la pestanya del càlcul plàstic passa a la funció
calcular2
        calcular2;
    end
    if get(botoCalculInestabilitat,'Value')
% Si està seleccionada la pestanya del càlcul d'inestabilitat passa a la funció
calcular3
        calcular3;
    end
end
```

Código 4.1. Función tipus_calcul

Esta función ha sido modificada para que las líneas de influencia sólo funcionen cuando estamos en la ventana de cálculo elástico. Entonces, si estamos en esta ventana, apretamos el botón de Linfluencia, y además le damos a Calcular, el programa calculará las líneas de influencia. Si en cambio, el botón Linfluencia no ha sido apretado, el botón Calcular hará las leyes de esfuerzos del cálculo estático.

```

function calcular_influencia
    % Surt del mode edició, elimina digrames anteriors.
    % Prepara les dades per cridar a la funcio de càlcul estàtic.
    % Representa les reaccions.
    % Amb el resultats del càlcul estàtic crida a la funció de càlcul i
    % dibuix dels diagrames i la deformada, els escala i col·loca al
    % seu lloc.

    % seccion: exemple 3 barres [0,3,0], la carga està aplicada en la
    % segona barra a 3 unitats del node

    % Definició variables noves
    barresinicial=barres;

    [seccion,Ncoord,~,~,~,~,~,~,~,~,barra_carrega_mom]=prepara_dades_calculinfluencia(contnus,nusos,contbarra,barresini
    cial); % Adapta les dades al format de la funció de càlcul.
    Sec=max(seccion);
    longT=0;
    diagramesinfluencia=diagramainfluencia; % Llista de diagrames (1 objecte diagrama per a cada barra).

    if grafica~=grafica_est % Si no s'esta presentant resultats de barra.
        clic_grafica_res
    end

    if contbarra % Si hi ha barres.

    [diagrames,diagramesinfluencia,hgreaccions]=esborra_diagramesireaccionsinfluencia(diagrames,diagramesinfluencia,hgr
    eaccions); % Esborra els diagrames i les reaccions anteriors si n'hi han.

    Tmax=0; % Tallant màxim.
    Mmax=0; % Moment màxim.
    Rmax=0; % Reacció màxima.

```

```

for p=1:length(barres) % Possa a zero totes les cargues
    barres(p).puntual(1)=0;
    barres(p).moment=zeros(1,5);
    barres(p).repartida=zeros(1,6);
    barres(p).projectada=zeros(1,6);
end

mm=linspace(0,long,10); % Punts equidistants on s'evaluarà

if mm(end)>Sec
    mm=[mm,Sec];
    mm=myunique(mm);
else
end

kk=1;
Reacciones=zeros(length(nusos),length(mm));
Posicion_reacciones=mm;

y=zeros(size(mm));
tt=[];

while kk<=length(mm)
    barres(ii).puntual(1)=1;
    barres(ii).puntual(5)=mm(kk);

[Ncoord,Ncond,Nextfor,Lnodes,Lartyp,Lmecp,barra_carrega_rep,barra_carrega_rep_proj,barra_carrega_punt,barra_carrega_mom]=prepara_dades_calcul(contnus,nusos,contbarra,barres); % Adapta les dades al format de la funcio de càlcul.
    [Retorn_reaccions,Retorn_moviments,Retorn_esforços_barres,TL2G,~,~,~,Vector_esforços_prescrits]
=
cme2_calcul_estatic(contnus,contbarra,Ncoord,Ncond,Nextfor,Lnodes,Lartyp,Lmecp,barra_carrega_rep,barra_carrega_rep_p
roj,barra_carrega_punt,barra_carrega_mom); % Crida a la funció de càlcul estàtic.

    [~,Ty,~,~,~,Mz]=calcul_tallant_moment_reaccions(seccion,ii); % Calcula les lleis del tallant i
del moment flector

```

```

% Calcula el valor del moment flector i el tallant en els punts equidistants
if Sec >= 0
    [m,~]=size(Mz);
    if m > 1
        if mm(kk) >= Sec
            y1=polyval(Mz(1,1:3),Sec);
        else
            y1=polyval(Mz(2,1:3),Sec);
        end
    else
        if isempty(Mz)==1
            y1=0;
        else
            y1=polyval(Mz(1,1:3),Sec);
        end
    end
    y(1,kk)=y1; % Valors moment flector

    [m,~]=size(Ty);
    tt2=[];
    if m > 1
        if mm(kk) > Sec
            tt1=polyval(Ty(1,1:2),Sec);
        elseif mm(kk) < Sec
            tt1=polyval(Ty(2,1:2),Sec);
        else
            tt1=polyval(Ty(2,1:2),Sec);
            tt2=polyval(Ty(1,1:2),Sec);
        end
    else
        if isempty(Ty)==1
            tt1=0;
        else
            tt1=polyval(Ty(1,1:2),Sec);
        end
    end
end
end

```

```

if Sec == 0    && ii<find(seccion==0)
    [m,~]=size(Ty);
    tt2=[];
    if m > 1
        tt1=polyval(Ty(2,1:2),long);
    else
        if isempty(Ty)==1
            tt1=0;
        else
            tt1=polyval(Ty(1,1:2),long);
        end
    end
end
tt=[tt,tt1,tt2]; % Valors tallant

% Calcula el valor de les reaccions en els punts equidistants
a=1;
M=[];
for l=1:length(nusos)
    M=[M;Retorn_reaccions(2*a,1)];
    a=a+3;
end

Reacciones(:,kk)=M;
barres(ii).puntual(1)=0;
barres(ii).puntual(5)=0;
kk=kk+1;
end

Mz=y; % Moment flector
Ty=tt; % Tallant
R=Reacciones; % Reaccions
Pr=Posicion_reacciones; % Posició reaccions

```

```

    % Calcula la posició del tallant
    if seccion(ii)< 0
        PrTy=Pr;
    else
        u=find(mm==Sec);
        if u~=1 && u~=length(mm)
            PrTy=[mm(1,1:u),mm(1,u:end)];
        else
            PrTy=Pr;
        end
    end

    [Ty]=correccio_tallant(ii,seccion,long,nusos,barres,Ty); % Corregeix el tallant, si es necessari,
quan la secció es un suport.

%FUNCION PRINCIPAL
[diagramesinfluencia(ii)]=dibuix_diagrames_reaccions(ii,long,Ty,Mz,R,Pr,PrTy,length(nusos));

set(diagramesinfluencia(ii).hgTy,'Parent',hg_diagrames)
set(diagramesinfluencia(ii).hgMz,'Parent',hg_diagrames)
set(diagramesinfluencia(ii).hgR,'Parent',hg_diagrames)

diagramesinfluencia(ii).num=ii; % Número de barra a la qual pertany el diagrama.

Tmax=max([Tmax,diagramesinfluencia(ii).Tmax]); % Calcula la relació tallant o axil/longitud màxima
de totes les barres.
Mmax=max([Mmax,diagramesinfluencia(ii).Mmax]); % Calcula la relació moment/longitud màxim de totes
les barres.
Rmax=max([Rmax,diagramesinfluencia(ii).Rmax]); % Calcula la relació moment/longitud màxim de totes
les barres.
longT=longT+long; % Suma de totes les barres
end
if Tmax<0.00001/long
    Tmax=1;
end
if Mmax<0.00001/long
    Mmax=1;
end
if Rmax<0.00001/long
    Rmax=1;
end
end

```

```

sT=(0.25*longT)/Tmax; % Càlcul de l'escala dels diagrames de tallants.
sM=(0.25*longT)/Mmax; % Càlcul de l'escala dels diagrames de moments.
sR=(0.25*longT)/Rmax; % Càlcul de l'escala dels diagrames de reaccions.

% Coloca, escala i coloreja els diagrames.
for i=1:numel(diagramesinfluencia)
    pos=nusos(barres(i).nusinf).posicio; % El diagrama es col·loca a la posicio del nus inferior.
    vector_barra=nusos(barres(i).nussup).posicio-pos; % Vector direcció de la barra, per a calcular la
matriu de rotacio dels diagrames. No s'utilitzen els eixos locals, porque poden no coincidir l'eix z local i
global.
    mT=matbarra(vector_barra,pos,sT); % Càlcul de la matriu de rotació, posició i escala dels diagrames
de tallant.
    mM=matbarra(vector_barra,pos,sM); % Càlcul de la matriu de rotació, posicio i escala dels diagrames
de moment.
    mR=matbarra(vector_barra,pos,sR); % Càlcul de la matriu de rotació, posicio i escala dels diagrames
de reaccions.

    set(diagramesinfluencia(i).hgTy,'Matrix',mT); % Col·locació del diagrama de tallant en y.
    set(diagramesinfluencia(i).hgMz,'Matrix',mM); % Col·locació del diagrama de moments en z.

    if strcmp(vista,'3d')
        clip3d='off'; % Fa que els objectes gràfics sobresurtin de la grafica.
    else
        clip3d='on'; % Fa que els objectes gràfics NO sobresurtin de la grafica.
    end

    set(get(diagramesinfluencia(i).hgTy,'Children'),'Clipping',clip3d)
    set(get(diagramesinfluencia(i).hgMz,'Children'),'Clipping',clip3d)

    for j=1:length(nusos)
        set(diagramesinfluencia(i).hgR(j),'Matrix',mR); % Col·locació del diagrama de reaccions.
        set(get(diagramesinfluencia(i).hgR(j),'Children'),'Clipping',clip3d)
    end

    colordigramareaccions(diagramesinfluencia(i))
end
end

```

```

numeroNusos=zeros(1,length(nusos));
for i=1:length(nusos)
    numeroNusos(1,i)=i;
end

for i=1:length(nusos)    % Oculta diagrama reaccions en nodes lliures.
    if nusos(i).contorn==0
        for j=1: numel(diagramesinfluencia)
            set(diagramesinfluencia(j).hgR(i),'Visible','off')
        end
    end
end

% Activació de les opcions del dibuix dels diagrames d'influència
set(bototallantYInf,'value',1);
set(botomomentZInf,'value',1);

% Visibilitat diagrames d'influencia
set(popupMenuInf,'Style','popupmenu','String',numeroNusos,'Position',[5,tm(2)-480,100,200],    'callback',
@popupcallbackinf); %Actualitza el popup
visibilitat_diagramesInf

% Recol·locació de l'estructura dibuixada i edició de la finestra
barres=barresinicial;
redibuixar_carregues_barres % Dibuixa una carga puntual en la secció d'estudi.
redibuixa_carreganusos
escalar_nusos
set(botoEdicio,'State','off'); % Botó edició en mode presentacio.
edicio % Desactiva botons de la barra d'eines per a l'edició, activa els de mostra de resultats, etc.

end

```

Código 4.2. Función calcular_influencia


```

function [seccion,Ncoord,Ncond,Nextfor,Lnodes,Lartyp,Lmecp,barra_carrega_rep,barra_carrega_rep_proj,barra_carrega_punt,barra_carrega_mom]=prepara_dades_calculinfluencia(contnus,nusos,contbarra,barres)
% Prepara les dades al format d'entrada de la funció de càlcul estàtic per calcular les línies d'influència.

Ncoord=zeros(contnus,3); % Coordenades dels nusos.
Ncond=zeros(contnus,7); % Condicions de contorn.
Nextfor=zeros(contnus*6,1); % Càrregues als nusos.
seccion=zeros(1,contbarra); % Secció on es troba la carga unitaria

for i=1:contnus
    Ncoord(i,:)=nusos(i).posicio; % Coordenades dels nusos.
    Ncond(i,1)=nusos(i).contorn; % Condicions de contorn.
    switch nusos(i).contorn
        case 1 % rodet
            Ncond(i,2:4)=nusos(i).rodet; % Condició de contorn rodet.
        case {4,5,6} % moviments elàstics
            Ncond(i,2:7)=nusos(i).rig; % Condició de contorn elàsticitat.
        case {7,8,9} % moviments imposats
            Ncond(i,2:7)=nusos(i).desp; % Condició de contorn moviment.
    end
    Nextfor(i*6-5:i*6-3)=nusos(i).puntual; % Càrregues puntuals als nusos.
    Nextfor(i*6-2:i*6)=nusos(i).moment; % Càrregues moment als nusos.
end

Lnodes=zeros(contbarra,2); % Connectivitat dels extrems de barra.
Lartyp=zeros(contbarra,2); % articulació dels extrems de barra.
Lmecp=zeros(contbarra,9); % Propietats mecàniques de les barres.
barra_carrega_rep=zeros(contbarra,6); % Càrregues repartides.
barra_carrega_rep_proj=zeros(contbarra,6); % Càrregues projectades.
barra_carrega_punt=zeros(contbarra,5); % Càrregues puntuals.
barra_carrega_mom=zeros(contbarra,5); % Càrregues moment.

for i=1:contbarra
    if barres(i).puntual(1)==0
        seccion(i) = -1;
    else
        seccion(i) = barres(i).puntual(5);
    end
end

```

```

for i=1:contbarra
    if barres(i).puntual(1)==0
        seccion(i) = -1;
    else
        seccion(i) = barres(i).puntual(5);
    end
    Lnodes(i,1)=barres(i).nusinf; % Connectivitat de l'extrem inferior de barra.
    Lnodes(i,2)=barres(i).nussup; % Connectivitat de l'extrem superior de barra.
    Lartyp(i,1)=barres(i).artinf; % articulacio de l'extrem inferior de barra.
    Lartyp(i,2)=barres(i).artsup; % articulacio de l'extrem superior de barra.
    Lmecp(i,1)=barres(i).E; % Modul de Young.
    Lmecp(i,2)=9.81; % Gravetat, herència de CME, de fet no s'utilitza.
    Lmecp(i,3)=barres(i).A; % Àrea de la secció.
    Lmecp(i,4)=barres(i).Iz; % Moment d'inèrcia Iz.
    Lmecp(i,5)=barres(i).Iy; % Moment d'inèrcia Iy.
    Lmecp(i,6:8)=barres(i).vectorIz; % Vector direcció inèrcia Iz.
    Lmecp(i,9)=barres(i).GIT; % Rigidesa a torsió.
    barra_carrega_rep(i,:)=barres(i).repartida; % Càrrega repartida.
    barra_carrega_rep_proj(i,:)=barres(i).projectada; % Càrrega projectada.
    barra_carrega_punt(i,:)=barres(i).puntual; % Càrrega puntual.
    barra_carrega_mom(i,:)=barres(i).moment; % Càrrega moment.
end
end

```

Código 4.2.1. Función prepara_dades_calculinfluencia

```

classdef diagramainfluencia
    %UNTITLED Summary of this class goes here
    % Detailed explanation goes here

    properties
        num=0; % Número de la barra a la qual pertanyen els diagrames.

        hgN=[]; % Handle de l'objecte hgtransform amb el diagrama d'axils.
        hgTy=[]; % Handle de l'objecte hgtransform amb el diagrama de tallants en y.
        hgTz=[]; % Handle de l'objecte hgtransform amb el diagrama de tallants en z.
        hgMx=[]; % Handle de l'objecte hgtransform amb el diagrama de moments en x.
        hgMy=[]; % Handle de l'objecte hgtransform amb el diagrama de momentss en y.
        hgMz=[]; % Handle de l'objecte hgtransform amb el diagrama de moments en z.

        hgR=[]; % Handle de l'objecte hgtransform amb el diagrama de reaaccions en z.

        Mmax=0; % Moment màxim absolut.
        Tmax=0; % Axil o tallant màxim absolut.
        Rmax=0; % Reacció màxima absoluta.

    end

    methods

    end

end

```

Código 4.2.2. Función diagramainfluencia

```

function [diagrames,diagramesinfluencia,hgreacciones]=esborra_diagramesireaccionesinfluencia(diagrames,diagrames
influencia,hgreacciones)
    % Esborra els diagrames i reaccions anteriors.

    for i=1:length(diagrames) % Esborra els diagrames anteriors.
        delete(diagrames(i).hgN)
        delete(diagrames(i).hgTy)
        delete(diagrames(i).hgTz)
        delete(diagrames(i).hgMx)
        delete(diagrames(i).hgMy)
        delete(diagrames(i).hgMz)
    end

    for i=1:length(diagramesinfluencia) % Esborra els diagrames anteriors.
        delete(diagramesinfluencia(i).hgN)
        delete(diagramesinfluencia(i).hgTy)
        delete(diagramesinfluencia(i).hgTz)
        delete(diagramesinfluencia(i).hgMx)
        delete(diagramesinfluencia(i).hgMy)
        delete(diagramesinfluencia(i).hgMz)
        delete(diagramesinfluencia(i).hgR)
    end

    for i=1:length(hgreacciones) % Esborra les reaccions anteriors.
        if hgreacciones(i)
            delete(hgreacciones(i))
        end
    end
    end
    hgreacciones=[];
end

```

Código 4.2.3. Función esborra_diagramesireaccionesinfluencia

```

function [N,Ty,Tz,Mx,My,Mz]=calcul_tallant_moment_reaccions(seccion,ii)
    N=0; % No tenim en compte els esforços que no interessin
    Tz=0;
    Mx=0;
    My=0;
    mov_extrems=zeros(12,contbarra);
    for i=1:contbarra % Calcula els esforços en la barra on estroba la secció d'estudi
        if seccion(i)>0
            [long,esforcos,~,rep,proj,punt,mom,~,~]=dades_diagramainfluencia(i);
            [~,Ty,~,~,~,Mz,diagrames(i)]=calcul_esforcos_influencia(seccion(i),long,esforcos,rep,proj,punt,mom); %
        Calcula i dibuixa els diagrames.
        elseif seccion(i)==0 && i==1
            [long,esforcos,~,rep,proj,punt,mom,~,~]=dades_diagramainfluencia(i);
            [~,Ty,~,~,~,Mz,diagrames(i)]=calcul_esforcos_influencia(seccion(i),long,esforcos,rep,proj,punt,mom); %
        Calcula i dibuixa els diagrames.
        elseif seccion(i)==0 && i~=1
            if ii<find(seccion==0)
                [long,esforcos,~,rep,proj,punt,mom,~,~]=dades_diagramainfluencia(i);
                [~,Ty,~,~,~,Mz,diagrames(i)]=calcul_esforcos_influencia(seccion(i),long,esforcos,rep,proj,punt,mom); %
            Calcula i dibuixa els diagrames.

            [long,esforcos,~,rep,proj,punt,mom,~,~]=dades_diagramainfluencia(i-1);
            [~,Ty,~,~,~,Mz]=calcul_esforcos_influencia(long,long,esforcos,rep,proj,punt,mom);
            else
                [long,esforcos,~,rep,proj,punt,mom,~,~]=dades_diagramainfluencia(i);
                [~,Ty,~,~,~,Mz,diagrames(i)]=calcul_esforcos_influencia(seccion(i),long,esforcos,rep,proj,punt,mom); %
            Calcula i dibuixa els diagrames.
        end
    end
end
end
end

```

Código 4.2.4 Función calcul_tallant_moment_reaccions

```

function
[N,Ty,Tz,Mx,My,Mz,diag,varargout]=calcul_esforcos_influencia(seccion,long,esforcos,repartida,projectada,puntual,moment)

detall=nargout-1; % Si s'hi introdueix més d'una sortida retorna els diagrames amb marques i dades numèriques per a la presentació en detall. Amb una sola sortida dibuixa els diagrames simples per a ser presentats a la grafica de l'estructura.
s=seccion;
sep=long/50; % Separació entre punts per al dibuix dels diagrames, 'resolució' del diagrama.
diag=diagrama; % Crea l'objecte de la classe diagrama.

[puntual,moment,repartida,projectada,dcx]=carregues(puntual,moment,repartida,projectada); % Normalitza les càrregues per al càlcul.

N=calcul_tallant(esforcos(1),[puntual(2),puntual(5)],[repartida(2),repartida(5),long-repartida(6)],[projectada(2),projectada(5),long-projectada(6)],long,dcx); % Expressió axils.
Ty=calcul_tallant(esforcos(2),[puntual(3),puntual(5)],[repartida(3),repartida(5),long-repartida(6)],[projectada(3),projectada(5),long-projectada(6)],long,dcx); % Expressió tallants en y.
Tz=calcul_tallant(esforcos(3),[puntual(4),puntual(5)],[repartida(4),repartida(5),long-repartida(6)],[projectada(4),projectada(5),long-projectada(6)],long,dcx); % Expressió tallants en z.

Mx=calcul_moment(0,esforcos(4),[moment(2),moment(5)],[0,0],[0,0,0],[0,0,0],long,dcx,1); % Expressió moment torsor.
My=calcul_moment(esforcos(3),esforcos(5),[moment(3),moment(5)],[puntual(4),puntual(5)],[repartida(4),repartida(5),long-repartida(6)],[projectada(4),projectada(5),long-projectada(6)],long,dcx,-1); % Expressió moment en y.
Mz=calcul_moment(esforcos(2),esforcos(6),[moment(4),moment(5)],[puntual(3),puntual(5)],[repartida(3),repartida(5),long-repartida(6)],[projectada(3),projectada(5),long-projectada(6)],long,dcx,1); % Expressió moment en z.

end

```

```

%% NORMALITZACIÓ DE LES CÀRREGUES.
%
% Multiplica el vector direcció de cada càrrega pel seu mòdul.
% Obté la rel·lació entre longitud de la càrrega projectada i de la barra.

function [puntual,moment,repartida,projectada,dcx]=carregues(puntual,moment,repartida,projectada)
    if puntual(1) && any(puntual(2:4))
        puntual(2:4)=puntual(1)*puntual(2:4)/norm(puntual(2:4));
    else
        puntual(2:4)=[0,0,0];
    end
    if moment(1) && any(moment(2:4))
        moment(2:4)=moment(1)*moment(2:4)/norm(moment(2:4));
    else
        moment(2:4)=[0,0,0];
    end
    if repartida(1) && any(repartida(2:4))
        repartida(2:4)=repartida(1)*repartida(2:4)/norm(repartida(2:4));
    else
        repartida(2:4)=[0,0,0];
    end
    if projectada(1) && any(projectada(2:4))
        projectada(2:4)=projectada(1)*projectada(2:4)/norm(projectada(2:4));
        vbarra=[1,0,0];
        vcarrega=projectada(2:4);
        vperp=cross(vbarra,vcarrega);
        vperp=cross(vperp,vcarrega);
        vperp=vperp/norm(vperp);
        dcx=abs(dot(vperp,vbarra));
    else
        projectada(2:4)=[0,0,0];
        dcx=0;
    end
end

```

```

%% CÀLCUL DELS POLINOMIS QUE DEFINEIXEN ELS DIAGRAMES D'AXILS I TALLANTS.

function T=calcul_tallant(forextrem,fpuntual,frepartida,fprojectada,long,dcx)
if forextrem || fpuntual(1) || frepartida(1) || fprojectada(1) % Si hi ha tallant en crea el diagrama.
    xT=[];
    % Defineix els trams del diagrama:
    if fpuntual(1)
        xT=fpuntual(2);
    end
    if frepartida(1)
        xT=[xT,frepartida(2),frepartida(3)];
    end
    if fprojectada(1)
        xT=[xT,fprojectada(2),fprojectada(3)];
    end
    xT=myunique([0,xT,long]);
    T=zeros(numel(xT)-1,4);
    % Per a cada tram calcula els coeficients de la expressió analítica del diagrama de tallants en y:
    for i=2:numel(xT)
        x0=xT(i-1);
        x1=xT(i);
        T(i-1,3:4)=[x0,x1];
        if fpuntual(1) && x1>fpuntual(2)
            T(i-1,1:2)=[0,-forextrem-fpuntual(1)];
        else
            T(i-1,1:2)=[0,-forextrem];
        end
        if frepartida(1) && x1>frepartida(2) && x1<=frepartida(3)
            T(i-1,1:2)=T(i-1,1:2)+[-frepartida(1),frepartida(1)*frepartida(2)];
        elseif frepartida(1) && x1>frepartida(3)
            T(i-1,1:2)=T(i-1,1:2)+[0,-frepartida(1)*(frepartida(3)-frepartida(2))];
        end
        if fprojectada(1) && x1>fprojectada(2) && x1<=fprojectada(3)
            T(i-1,1:2)=T(i-1,1:2)+[-fprojectada(1)*dcx,fprojectada(1)*fprojectada(2)*dcx];
        elseif fprojectada(1) && x1>fprojectada(3)
            T(i-1,1:2)=T(i-1,1:2)+[0,-fprojectada(1)*(fprojectada(3)-fprojectada(2))*dcx];
        end
    end
else
    T=[];
end
end

```



```

%% CÀLCUL DELS POLINOMIS QUE DEFINEIXEN ELS DIAGRAMES DE MOMENTS.

function M=calcul_moment(forextrem,momextrem,fmoment,fpuntual,frepartida,fprojectada,long,dcx,signe)
    if forextrem || momextrem || fmoment(1) || fpuntual(1) || frepartida(1) || fprojectada(1) % Si hi ha moments en
eix z en crea el diagrama.
        xM=[];
        if fmoment(1)
            xM=fmoment(2);
        end
        if fpuntual(1)
            xM=[xM,fpuntual(2)];
        end
        if frepartida(1)
            xM=[xM,frepartida(2),frepartida(3)];
        end
        if fprojectada(1)
            xM=[xM,fprojectada(2),fprojectada(3)];
        end
        xM=myunique([0,xM,long]);
        M=zeros(numel(xM)-1,5);
        for i=2:numel(xM)
            x0=xM(i-1); % Valor X a l'inici del tram.
            x1=xM(i); % Valor X al final del tram.
            M(i-1,4:5)=[x0,x1]; % Afegeix les dades d'X inicial i final a cada expressió analítica.

            % Expressió analítica del diagrama de moments segons els esforços.
            % En forma de vector amb els valors dels factors.
            if fmoment(1) && xM(i)>fmoment(2)
                M(i-1,1:3)=[0,forextrem*signe,-momextrem-fmoment(1)];
            else
                M(i-1,1:3)=[0,forextrem*signe,-momextrem];
            end
            if fpuntual(1) && x1>fpuntual(2)
                M(i-1,1:3)=M(i-1,1:3)+[0,fpuntual(1),-fpuntual(1)*fpuntual(2)]*signe;
            end
        end
    end
end

```

```

        if frepartida(1) && x1>frepartida(2) && x1<=frepartida(3)
            M(i-1,1:3)=M(i-1,1:3)+[frepartida(1)/2,-
frepartida(1)*frepartida(2),frepartida(1)*frepartida(2)^2/2]*signe;
        elseif frepartida(1) && x1>frepartida(3)
            M(i-1,1:3)=M(i-1,1:3)+[0,frepartida(1)*(frepartida(3)-frepartida(2)),-frepartida(1)*(frepartida(3)-
frepartida(2))*(frepartida(3)+frepartida(2))/2]*signe;
        end
        if fprojectada(1) && x1>fprojectada(2) && x1<=fprojectada(3)
            M(i-1,1:3)=M(i-1,1:3)+[fprojectada(1)*dcx/2,-
fprojectada(1)*fprojectada(2)*dcx,fprojectada(1)*dcx*(fprojectada(2))^2/2]*signe;
        elseif fprojectada(1) && x1>fprojectada(3)
            M(i-1,1:3)=M(i-1,1:3)+[0,fprojectada(1)*(fprojectada(3)-fprojectada(2))*dcx,-
fprojectada(1)*dcx*(fprojectada(3)-fprojectada(2))*(fprojectada(3)+fprojectada(2))/2]*signe;
        end
    end
else
    M=[];
end
end
end

```

Código 4.2.4.1 Función calcul_esforços_influencia

```

function [long,esforcos,moviments,rep,proj,punt,mom,Iz,Iy]=dades_diagramainfluencia(n)
    n1=barres(n).nusinf;
    n2=barres(n).nussup;
    T=TL2G(:, :, n)'; % Matriu de canvi de base d'exos globals a locals.
    esforcos(1:12)=Retorn_esforcos_barres(n,1:12); % Esforços en extrems de barra en eixos locals.
    moviments=[T*Retorn_moviments(n1*6-5:n1*6-3);T*Retorn_moviments(n1*6-2:n1*6);T*Retorn_moviments(n2*6-
5:n2*6-3);T*Retorn_moviments(n2*6-2:n2*6)]; % Moviments dels extrems de barra en eixos locals.
    mov_extrems(:,n)=moviments; % Guarda els moviments en extrems de barra en eixos locals per a la funcio que
mostra l'informe de resultats.
    vbarra=nusos(n2).posicio-nusos(n1).posicio; % Vector barra.
    long=norm(vbarra); % Longitud de la barra.
    rep=[barres(n).repartida(1), (T*barres(n).repartida(2:4)')',barres(n).repartida(5:6)]; % Carrega repartida
en eixos locals.
    proj=[barres(n).projectada(1), (T*barres(n).projectada(2:4)')',barres(n).projectada(5:6)]; % Càrrega
projectada en eixos locals.
    punt=[barres(n).puntual(1), (T*barres(n).puntual(2:4)')',barres(n).puntual(5)]; % Carrega en eixos locals.
    mom=[barres(n).moment(1), (T*barres(n).moment(2:4)')',barres(n).moment(5)]; % Moments en eixos locals.
    Iz=barres(n).Iz; % Inercia Iz.
    if all(barres(n).vectorIz==0) % Si no esta definit el vector direcció d'Iz, Iy=Iz.
        Iy=Iz;
    else
        Iy=barres(n).Iy;
    end
end

```

Código 4.2.4.2 Función dades_diagramainfluencia

```

function [Ty]=correccio_tallant(ii,seccion,long,nusos,barres,Ty)
% Possa el tallant correcte quan la secció està a sobre d'un suport (Sec=0)
Tyinicial=Ty;
    if ii<length(seccion) && seccion(ii+1)==0 && find(seccion==0)==(ii+1)
        if Ty(end-1)> 1e-5
            Ty(end)=1;
        elseif Ty(end-1) < -1e-5
            Ty(end)=-1;
        else
            Ty(end)=0;
        end
    end

    if seccion(ii)==0
        if find(seccion==0)==(ii+1)
            if Ty(end-1)> 1e-5
                Ty(end)=1;
            elseif Ty(end-1) < -1e-5
                Ty(end)=-1;
            else
                Ty(end)=0;
            end
        elseif find(seccion==0)==ii
            if Ty(2)> 1e-5
                Ty(1)=1;
            elseif Ty(2) < -1e-5
                Ty(1)=-1;
            else
                Ty(1)=0;
            end
        end
    end
end

```

```

    if ii==find(seccion==long)
        if Ty(end-1)> 1e-5
            Ty(end)=1;
        elseif Ty(end-1) < -1e-5
            Ty(end)=-1;
        else
            Ty(end)=0;
        end
    end
    if nusos(ii).contorn==0 % Anul·la el tallant si la secció està en el extrems de barra d'un node lliure
        if seccion(ii)==0 && ii==1
            Ty(1)=0;
        end
        if seccion(ii)==0 && ii==length(barres)
            Ty(end)=0;
        end
    end
    if nusos(ii).contorn~=0 % Pone el cortante si la seccion está en los extremos de barra de un apoyo
        if seccion(ii)==0 && ii==1
            if Ty(2)>0
                Ty(1)=1;
            else
                Ty(1)=-1;
            end
        end
        if seccion(ii)==0 && ii==length(barres)
            if Ty(end-1)>0
                Ty(1)=1;
            else
                Ty(1)=-1;
            end
        end
    end
    if Ty==Tyinicial
        Ty=Tyinicial;
    else
        end
end
end

```

Código 4.2.5 Función correccio_tallant

```

function [diag]=dibuix_diagrames_reaccions(ii,long,Ty,Mz,R,Pr,PrTy,nudos)
diag=diagramainfluencia; % Crea l'objecte de la classe diagrama.
    for i=1:nudos
        [diag.hgR(i),Rmin(i,:),Rmax(i,.)]=dibudiag_reaccions_moment('R',R(i,:),Pr,long,1,1); % Dibuixa el
diagrama de reaccions en z.
        diag.Rmax(i)=max(abs([Rmin(i,2),Rmax(i,2)])); % Màxima reacció en termes absoluts, per a determinar
l'escala dels diagrames.
    end

    [diag.hgTy,Tymin,Tymax]=dibudiag_tallant('Ty',Ty,PrTy,long,1,1); % Dibuixa el diagrama de tallants en y.
    [diag.hgMz,Mzmin,Mzmax]=dibudiag_reaccions_moment('Mz',Mz,Pr,long,1,1); % Dibuixa el diagrama de moments en z.
    diag.Tmax=max(abs([Tymin(2),Tymax(2)])); % Màxim tallant o axil en termes absoluts, per a determinar l'escala
dels diagrames.
    diag.Mmax=max(abs([Mzmin(2),Mzmax(2)])); % Màxim moment en termes absoluts, per a determinar l'escala dels
diagrames.

end

```

Código 4.2.6 Función dibujar_diagramas_reacciones

```

function [hg,miny,maxy]=dibudiag_reaccions_moment(txt,P,coordP,long,detall,signe)
if ~isempty(P)
    hg=hgtransform;
    miny=[0,Inf];
    maxy=[0,-Inf];
    [n,l]=size(P);
    xd=0;
    yd=0;

    for i=1:n % En cada tram on es defineix el diagrama...
        x=coordP; % Coordenades dels punts equidistants.
        y=P; % Valor evaluat en els punts.
        xd=[xd,x]; % Afegeix els punts al diagrama.
        yd=[yd,y];
        if abs(sum(y)) < 0.000001
            y=zeros(size(y));
        end
        [miny(2),k]=min([miny(2),y]); % Comproba si el mínim i màxim del tram són els absoluts.
        if k~=1
            miny(1)=x(k-1);
        end
        [maxy(2),k]=max([maxy(2),y]);
        if k~=1
            maxy(1)=x(k-1);
        end
    end

    xd=[xd,coordP(n,l)]; % Uneix el punt final del diagrama amb la barra.
    yd=[yd,0];
    line(xd,-yd*signe,'Linewidth',2,'Parent',hg,'HitTest','off','Clipping','on') % Dibuixa el diagrama.
end

```

```

    if detall
        if miny(1)~=0 && miny(1)~=long
            scatter(miny(1),-miny(2)*signe,100,'+','k','Parent',hg) % Marca el mínim absolut.
            txtdiag(miny(1),miny(2),txt,long,hg,signe)
        end
        if maxy(1)~=0 && maxy(1)~=long && maxy(1)~=miny(1)
            scatter(maxy(1),-maxy(2)*signe,100,'+','k','Parent',hg) % Marca el màxim absolut.
            txtdiag(maxy(1),maxy(2),txt,long,hg,signe)
        end
        txtdiag(0,yd(2),txt,long,hg,signe) % Moment inicial.
        txtdiag(long,yd(length(yd)-1),txt,long,hg,signe) % Moment final.
    end
else
    hg=[];
    miny=[0,0];
    maxy=[0,0];
end
end

```

Código 4.2.6.1 Función dibudiag_reaccions_moment

```

function txtdiag(x,y,txt,long,hg,signe)
    if x>long/2
        halin='right';
    else
        halin='left';
    end
    if y>0
        valin='top';
    else
        valin='bottom';
    end
    text(x,-
y*signe,[txt,'(x=',num2str(x),')=',num2str(y)],'VerticalAlignment',valin,'HorizontalAlignment',halin,'Parent',hg);
end

```

Código 4.2.6.1.1 Función txtdiag


```

function [hg,miny,maxy]=dibudiag_tallant(txt,P,coordP,long,detall,signe)
    if ~isempty(P)
        hg=hgtransform;
        miny=[0,Inf];
        maxy=[0,-Inf];
        [n,l]=size(P);
        xd=0;
        yd=0;
        for i=1:n % En cada tram on es defineix el diagrama...
            x=coordP; % Ordena els punts i en descarta els repetits.
            y=P; % Evalua els punts.
            xd=[xd,x]; % Afegeix els punts al diagrama.
            yd=[yd,y];
            [miny(2),k]=min([miny(2),y]); % Comproba si el mínim i màxim del tram són els absoluts.
            if k~=1
                miny(1)=x(k-1);
            end

            if abs(miny(2)) < 0.0000001
                miny(1)=0;
            end

            [maxy(2),k]=max([maxy(2),y]);
            if k~=1
                maxy(1)=x(k-1);
            end

            if abs(maxy(2)) < 0.0000001
                maxy(1)=0;
            end
        end

        xd=[xd,coordP(n,l)]; % Uneix el punt final del diagrama amb la barra.
        yd=[yd,0];
        line(xd,-yd*signe,'Linewidth',2,'Parent',hg,'HitTest','off','Clipping','on') % Dibuixa el diagrama.
    end
end

```

```

    if detall
        if miny(1)~=0 && miny(1)~=long
            scatter(miny(1),-miny(2)*signe,100,'+','k','Parent',hg) % Marca el mínim absolut.
            txtdiag(miny(1),miny(2),txt,long,hg,signe)
        end
        if maxy(1)~=0 && maxy(1)~=long && maxy(1)~=miny(1)
            scatter(maxy(1),-maxy(2)*signe,100,'+','k','Parent',hg) % Marca el màxim absolut.
            txtdiag(maxy(1),maxy(2),txt,long,hg,signe)
        end
        if maxy(1)~=0 && maxy(1)~=long && maxy(1)==miny(1)
            scatter(maxy(1),-maxy(2)*signe,100,'+','k','Parent',hg) % Marca el màxim absolut.
            txtdiag(maxy(1),maxy(2),txt,long,hg,signe)
        end

        txtdiag(0,yd(2),txt,long,hg,signe) % Tallant inicial.
        txtdiag(long,yd(length(yd)-1),txt,long,hg,signe) % Tallant final.
    end
else
    hg=[];
    miny=[0,0];
    maxy=[0,0];
end
end
end

```

Código 4.2.6.2 Función dibudiag_tallant

```

function colordiagrama_influencia(diag)
    % Coloreja els digramaes i la deformada de l'objecte diagrama.
    chgcolor(diag.hgTy,colorTy); % Color diagrama de tallants en y.
    chgcolor(diag.hgMz,colorMz); % Color diagrama de moments en z.
    chgcolor(diag.hgR,colorN); % Color diagrama de reaccions en z.
end

```

Código 4.2.2.7 Función colordiagrama_influencia

```

function chgcolor(h,color)
    % Funció que canvia els colors dels objectes continguts en un
    % objecte hgtransform.

    set(findobj(h,'Type','line'),'color',color) % Color de les línies.
    set(findobj(h,'Type','text'),'color',color) % Color dels textos.
    set(findobj(h,'Type','patch'),'FaceColor',color,'EdgeColor',color) % Color dels patch.
end

```

Código 4.2.2.7.1 Función chgcolor

4.2.1. Función *prepara_dades_calculinfluencia*

Esta función prepara la información introducida en el programa para calcular las líneas de influencia. Es la misma función que ya había en el programa pero se ha modificado para introducir una nueva variable llamada sección.

La variable “sección” es un vector de una fila y tantas columnas como barras tengamos. Si tenemos una viga de tres vanos de 2, 1 y 1.5 metros, la variable sección será del tipo [1 -1 -1], si la carga está en el primer vano y a una distancia de 1 metro del primer nodo, [-1 0.75 -1], si la carga está en el segundo vano y a una distancia de 0.75 del segundo nodo, o bien [-1 -1 1.5] si la carga está en el tercer vano y apoyada en el tercer nodo. La variable funciona de manera muy sencilla. Nosotros queremos calcular las líneas de influencia en un punto, por lo tanto, dibujaremos una carga, o bien en un apoyo, o bien en un punto de la barra. Veámoslo en un ejemplo:

Siguiendo con un ejemplo similar al anterior, si nosotros tenemos una viga de tres vanos de 1, 2 y 3 metros y queremos conocer las líneas de influencia en el centro del segundo vano sólo tenemos que dibujar nuestra carga puntual en esa sección y, una vez demos al botón calcular se creará la variable “sección”. En este caso, sección = [-1 1 -1]. Los -1 indican que la carga no está aplicada ni en la barra primera ni en la tercera barra. El número positivo 1 indica que la carga está aplicada en la segunda barra y a una distancia de 1 metro del nudo inicial de esa barra, que por defecto, siempre será el nudo izquierdo, en este caso nudo 2.

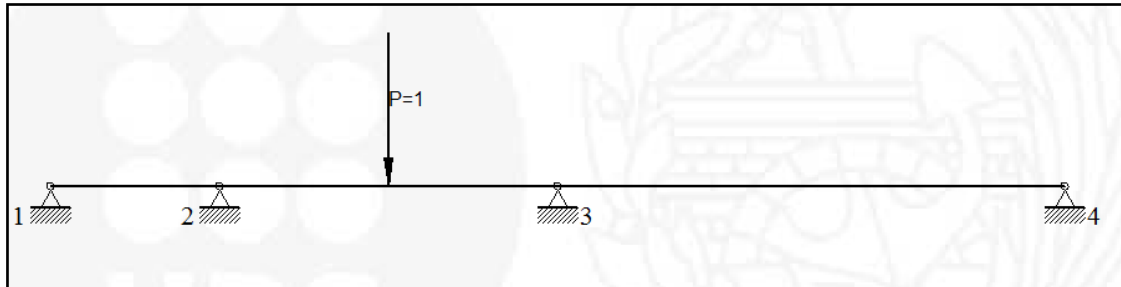


Figura 20. Viga ejemplo para explicar la variable sección

La variable “sección” nos servirá para saber cuál es la sección donde queremos conocer las líneas de influencia, es decir, la sección de estudio.

4.2.2. Función *diagramainfluencia*

Introduce un nuevo objeto para guardar los diagramas de influencia.

4.2.3. Función *esborra_diagramesireaccionsinfluencia*

Esta función elimina los diagramas del cálculo estático y los diagramas de influencia anteriores, en el caso de que los haya, para que no haya problemas.

4.2.4. Función *calcul_tallant_moment_reaccions*

En primer lugar pone a cero las variables que no van a ser calculadas, ya que en este trabajo sólo vamos a tener en cuenta las reacciones, el cortante y el momento flector para vigas bidimensionales, por ese motivo, el axil, el cortante y el momento flector en los otros planos son igualados a cero.

Esta función calcula la ley de esfuerzos del cortante y del momento flector llamando a la función *calcul_esforcos_influencia*. Ésta toma como referencia la sección en donde está aplicada nuestra carga unitaria. Va mirando barra por barra y si nuestra carga está aplicada en la barra de estudio entonces calcula los esfuerzos sino no calcula nada.

Ejemplo:

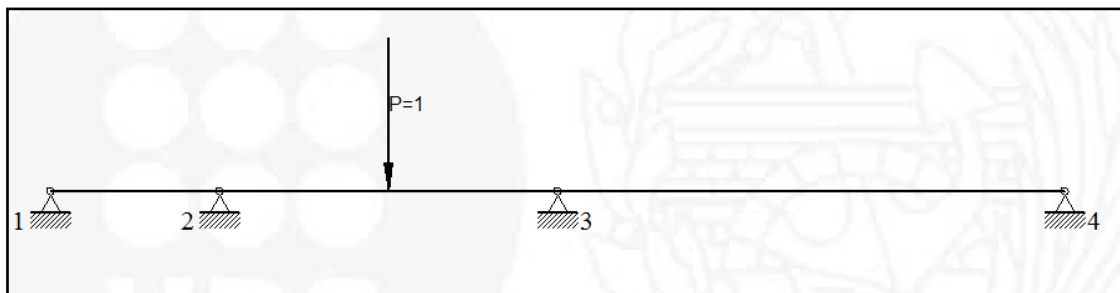


Figura 21. Ejemplo viga de tres vanos

Usaremos el ejemplo anterior. El programa empezaría calculando los puntos a evaluar de la barra 1 y entonces pasaríamos al cálculo de los esfuerzos donde la sección de estudio está en la barra 2, sección = [-1 1 -1]. Estos puntos equidistantes van desde el nodo 1 al nodo 2 y serán evaluados sólo en la sección donde está aplicada la carga, es decir, en la mitad de la segunda barra. Podemos observar que sólo se calcularán los esfuerzos cuando $seccion(i) > 0$ o $seccion(i) = 0$, es decir, cuando el número es no negativo o lo que es lo mismo, los esfuerzos se calcularán en la barra donde está aplicada la carga. Sabiendo que i es el subíndice de las barras, esto solo se cumple cuando la barra es la segunda. La función ignorará las barras 1-2 y 3-4. Después pasaríamos a la barra 2 y sucedería lo mismo, se ignorarán las barras 1-2 y 3-4. Y finalmente en la barra 3 igual que con las otras dos.

Se hace la distinción entre $seccion(i) > 0$ y $seccion(i) = 0$ porque, aunque el momento flector es el mismo se mire por la derecha que por la izquierda, el cortante cambia de sentido y dicho motivo se hacen varias distinciones en la función. Veámoslas:

Caso 1. sección(i) > 0

Sería el caso anterior, donde la carga está aplicada en algún punto que no sea el primer nodo de una barra.

Caso 2. sección(i) = 0

- a) Si la carga está aplicada en el primer nodo de la primera barra (Figura 22).

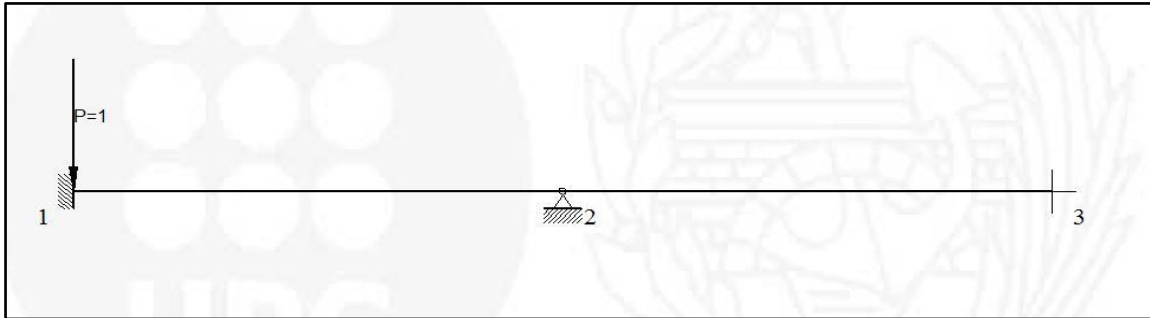


Figura 22. Ejemplo viga con carga aplicada en el nodo inicial

- b) Si la carga está aplicada en cualquier otro primer nodo que no sea el de la primera barra. Sólo se tiene en cuenta si la carga queda a la derecha de la barra de estudio (Figura 23).



Figura 23. Ejemplo viga con carga aplicada en un apoyo intermedio

Si la barra de estudio es la primera, segunda o tercera, se calculará de esta forma y, cuando se llegue a la cuarta barra se calcularán las leyes como en el caso 1.

4.2.4.1. Función calcul_esforcos_influencia

Prepara y calcula algunas variables para poder usar la función calcul_esforcos_influencia. Calcula la longitud de la barra del loop actual, los esfuerzos en los extremos de barra, los movimientos y las cargas dibujadas. Esta función es la misma que dades_diagrama ya incluida en el programa.

4.2.4.2. Función *dades_diagramainfluencia*

Es la misma función que la usada en *cme2_calcul_estatic* pero más sencilla y simplificada, ya que únicamente calcula las leyes de los esfuerzos y no crea ningún diagrama.

4.2.5. Función *correccio_tallant*

Como se ha mencionado anteriormente, las leyes de esfuerzos se evalúan en unos puntos equidistantes y estos puntos contienen tanto el primer nodo como el último de cada barra. En el caso del momento flector, en estos puntos inicial y final, no hay ningún problema dado que una carga situada en un apoyo da momento flector nulo. Pero sucede lo mismo si usamos la función anterior *calcul_esforcos_influencia*, ya que una carga situada en un apoyo no da cortante y esto, no siempre es cierto. Por ese motivo, se corrigen esos dos puntos conflictivos de los apoyos para que el diagrama quede correcto. Veamos todos los casos posibles:

Condición 1: La sección de estudio está en el último nodo de cualquier barra de estudio salvo la última barra

En la siguiente figura podemos ver un ejemplo de este caso en la barra 1.



Figura 24. Ejemplo viga de la condición 1

Si no hicieramos la corrección del cortante la línea de influencia quedaría de la siguiente manera (Figura 26).

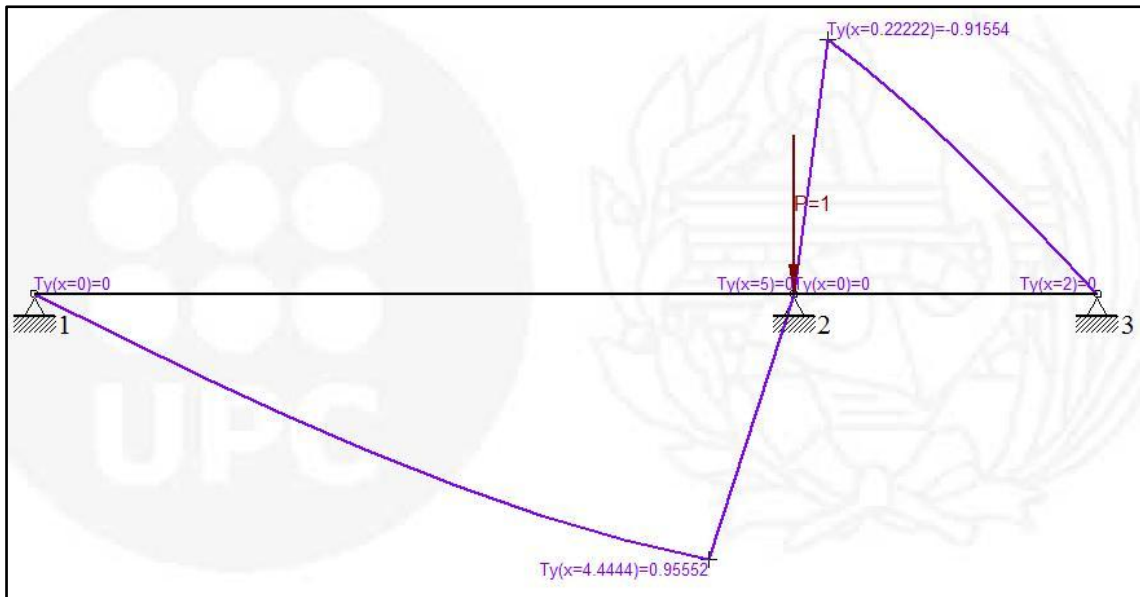


Figura 25. Ejemplo línea de influencia del cortante de la condición 1 sin corregir

Valores del cortante en 10 puntos equidistantes de la primera barra (nodos 1-2)

Debido a que el cortante en los puntos equidistantes es el siguiente:

0,0000	0,1503	0,2977	0,4392	0,5718	0,6927	0,7989	0,8875	0,9555	0,0000
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Debido a que el cortante en los puntos equidistantes es el mostrado en la tabla anterior, la función detecta que el cortante crece desde 0 hasta 1, en el caso de la primera barra y, decrece desde 0 hasta -1, en la segunda barra de derecha a izquierda.

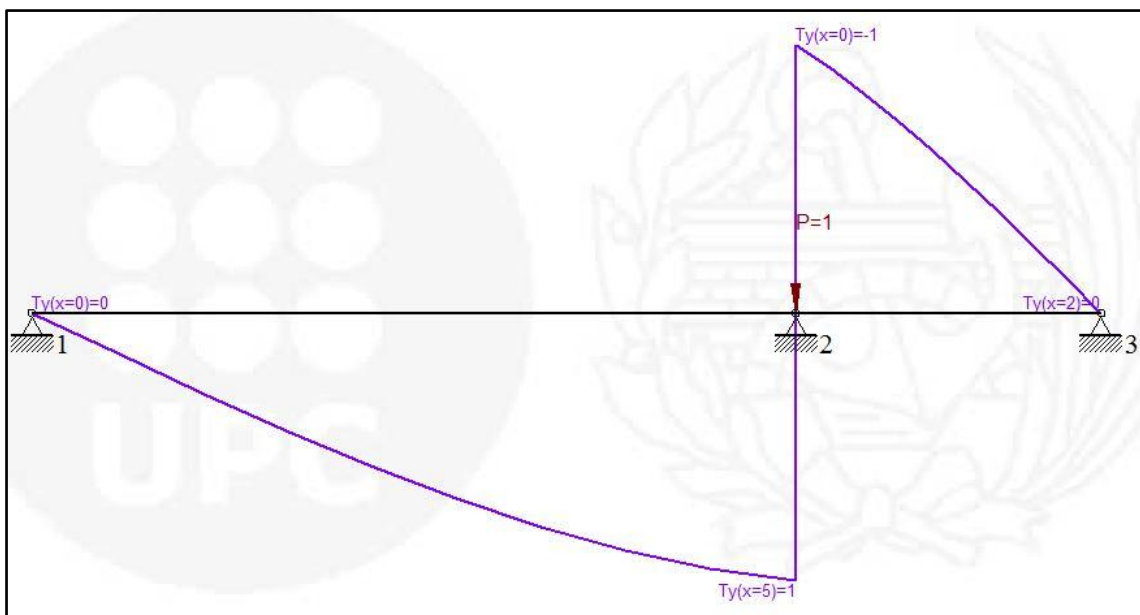


Figura 26 . Ejemplo línea de influencia del cortante de la condición 1 corregida

Condición 2. La sección de estudio está situada en el primer nodo de la barra de estudio

Un ejemplo de esta condición es el mismo que el de la barra anterior solo que, esta vez, la barra de estudio es la barra 2. En este caso pasaría lo mismo que lo explicado anteriormente y podemos ver cómo quedaría la línea de influencia del cortante en la figura.

Condición 3. La sección de estudio está situada en el último nodo de la última barra de estudio



Figura 27. Ejemplo viga de la condición 3

Podemos ver un ejemplo de esta condición en la Figura 27 . Del mismo modo que en los casos anteriores si no usamos la función corregida, se dibujaría una línea de influencia como la de la figura 28. Podemos ver la solución final al ejecutar la función `correccio_tallant` (Figura 29).

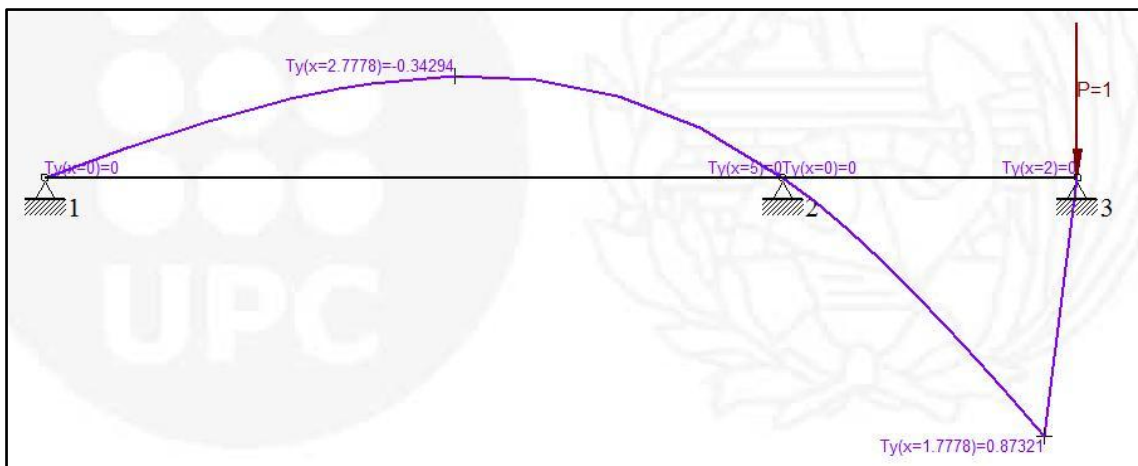


Figura 28. Ejemplo línea de influencia del cortante de la condición 3 sin corregir

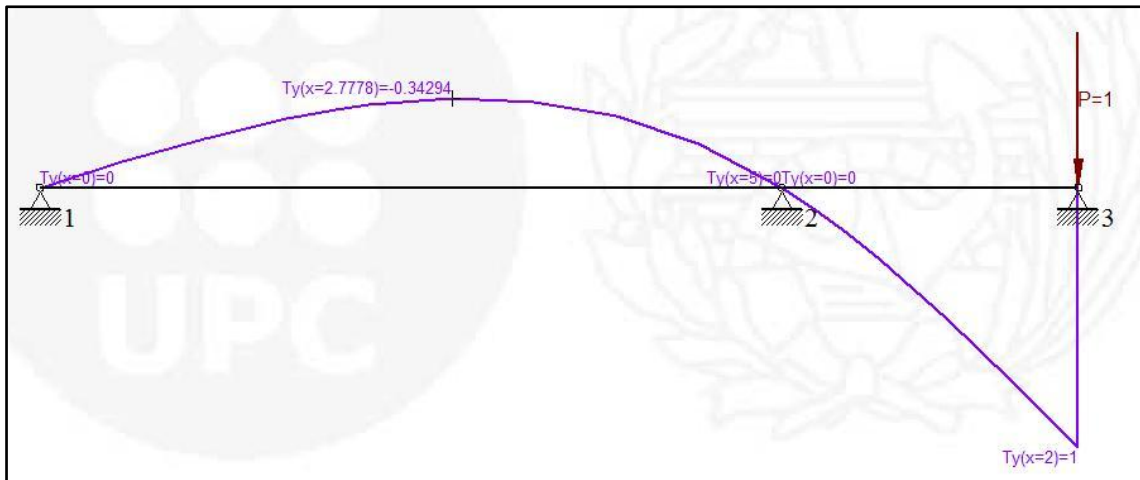


Figura 29. Ejemplo línea de influencia del cortante de la condición 1 corregida

Condición 4. La sección de estudio está situada en un extremo libre



Figura 30. Ejemplo 1 de viga de la condición 4

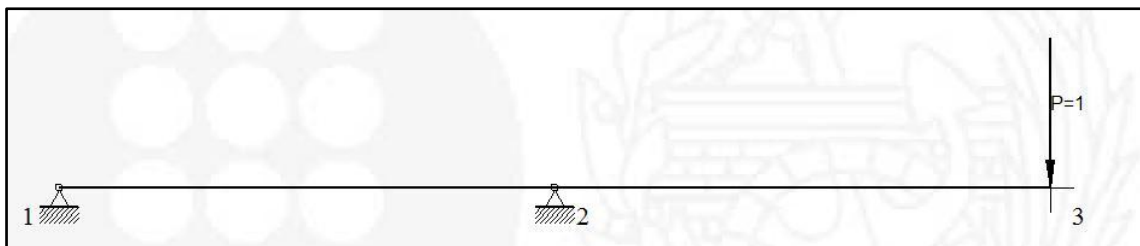


Figura 31. Ejemplo 2 de viga de la condición 4

En ambos casos el cortante es nulo y coincide con los valores de las leyes de momentos. En las figuras 30 y 31 podemos ver dos ejemplos de este caso.

4.2.6. Función `dibuix_diagrames_reaccions_moment`

Para cada barra dibuja las líneas de influencia usando las funciones `dibugdiag_reaccions_moment` y `txtdiag`.

4.2.6.1 Función dibudiag_reaccions_moment

Al ejecutar esta función, se relacionan la posición de los puntos equidistantes con los resultados obtenidos y se añaden más puntos en el caso de que sean necesarios. También se buscan los máximos y mínimos del diagrama y se procede a dibujarlo bajo la variable hg. Esta función es la encargada de dibujar las líneas de influencia en los apoyos y también la línea de influencia del momento flector en el eje Z.

4.2.6.1.1. Función txtdiag

Crea el texto de las leyes de esfuerzos con su valor numérico y lo posiciona en el diagrama.

4.2.6.2. Función dibudiag_tallant

La ejecución de esta función es similar a la anterior sólo que en vez de dibujar las líneas de influencia de las reacciones y del momento flector dibuja la línea de influencia del cortante.

4.2.7. Función colordiagramareaccions

Con esta función se da color a los diagramas de influencia. Los colores usados para el cortante y el momento flector son los mismos que en el cálculo estático. Y para el diagrama de influencia de las reacciones se ha usado el mismo color que el de los esfuerzos axiales en el cálculo estático.

CAPÍTULO 3

1. Ejemplos de líneas de influencia

1.1. Viga biapoyada

Ejemplo 1. Viga biapoyada de 10 metros con sección de estudio a 8 metros del primer apoyo

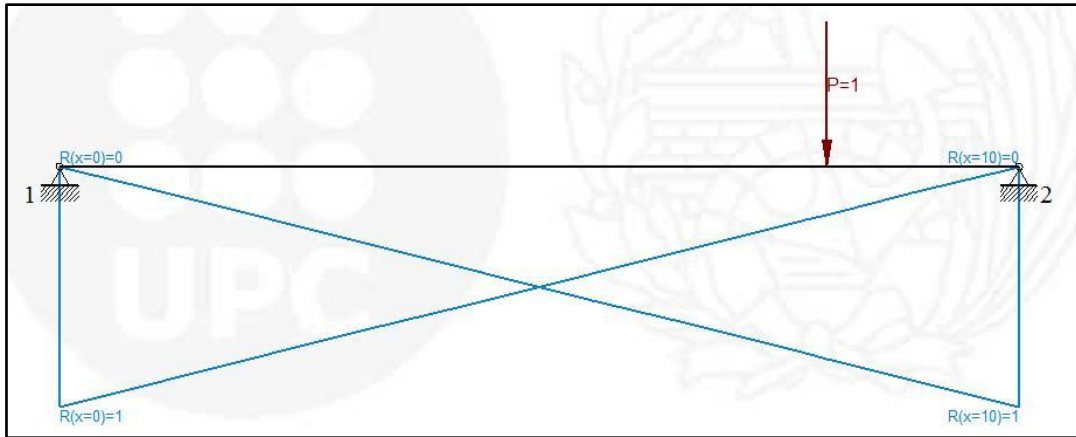


Figura 32. Líneas de influencia de las reacciones en los apoyos 1 y 2



Figura 33. Línea de influencia del cortante en la sección P

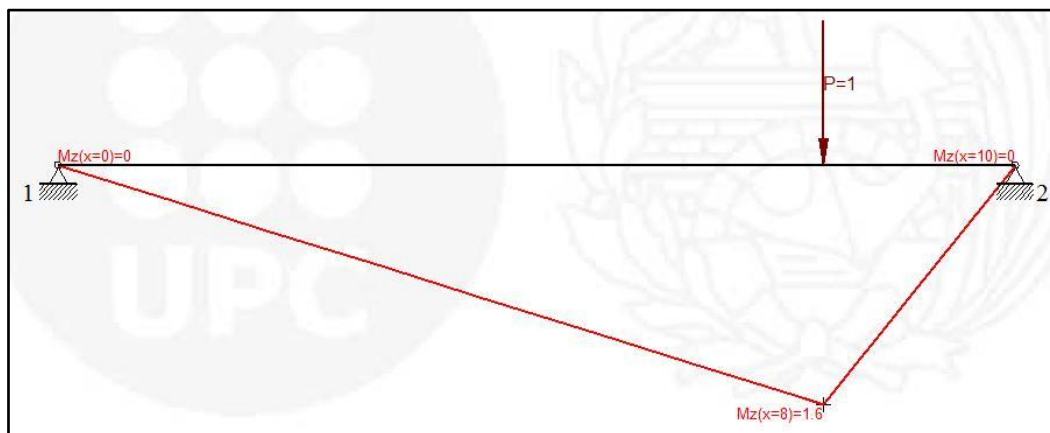


Figura 34. Línea de influencia del momento flector en la sección P

1.2. Viga biapoyada con voladizo

Ejemplo 2. Viga biapoyada con voladizo de 10 metros y sección a 5 metros del segundo apoyo

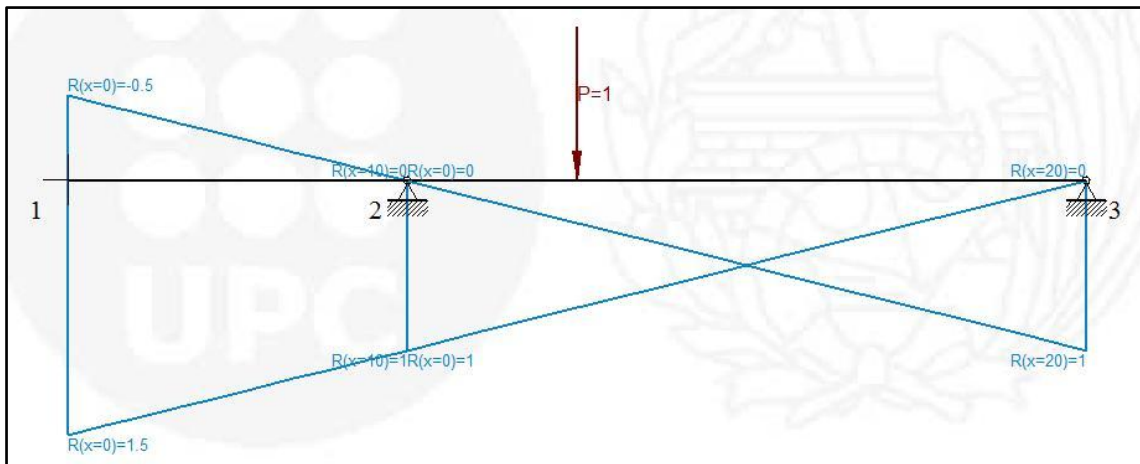


Figura 35. Líneas de influencia de las reacciones en los apoyos 2 y 3

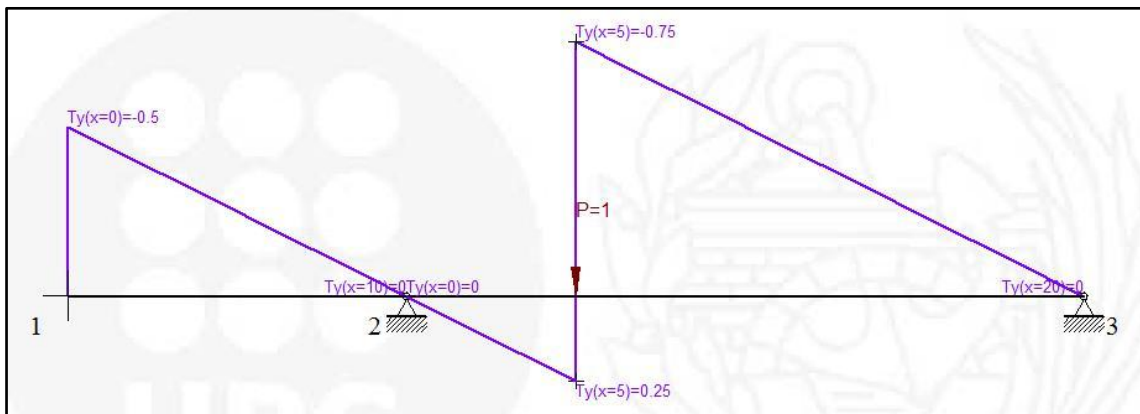


Figura 36. Línea de influencia del cortante en la sección P

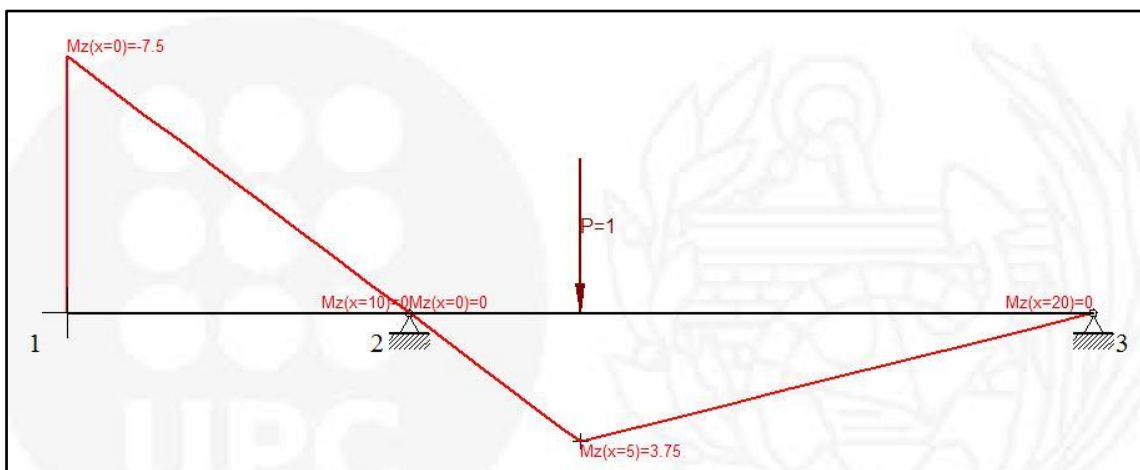


Figura 37. Línea de influencia del momento flector en la sección P

1.3. *Viga con un apoyo y un empotramiento*

Ejemplo 3. Viga apoyada y empotrada de 4 metros con sección de estudio a 2 metros

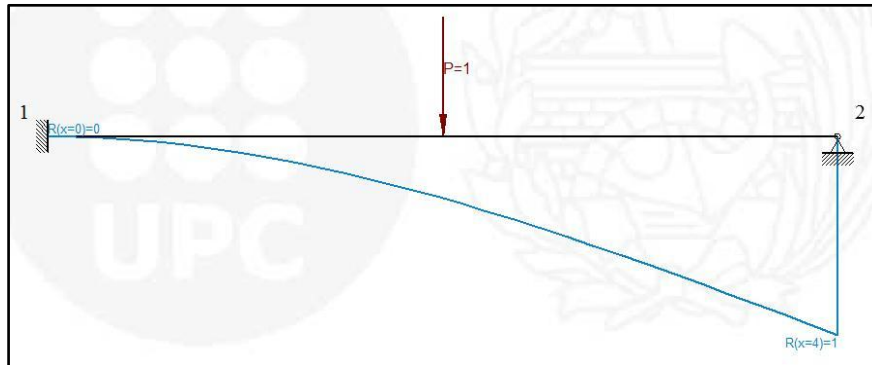


Figura 38. Línea de influencia de la reacción en el apoyo 1

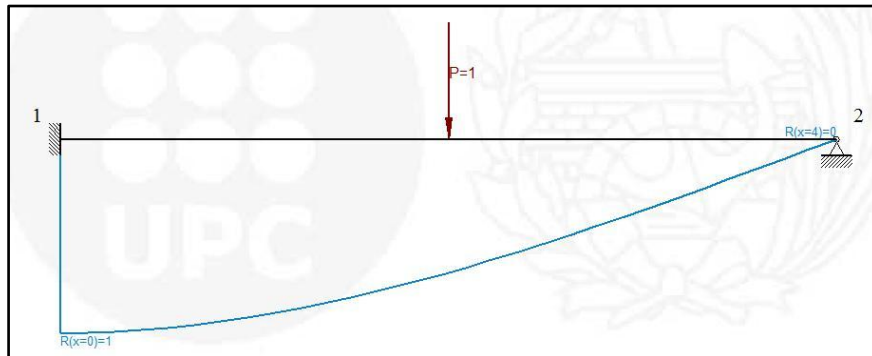


Figura 39. Línea de influencia de la reacción en el apoyo 2

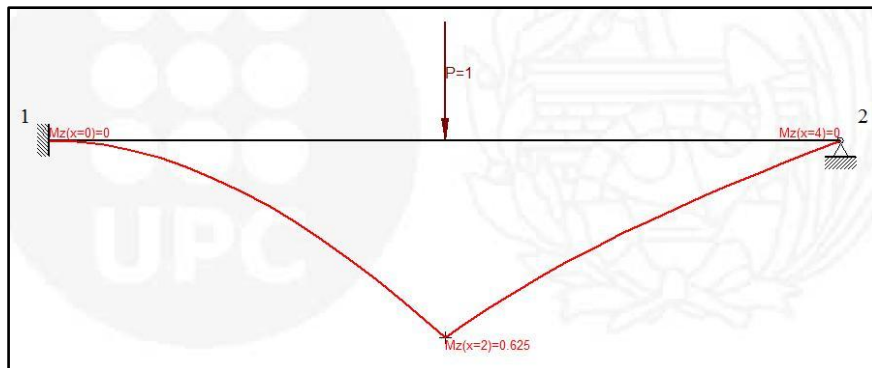


Figura 40. Línea de influencia del momento flector en la sección P

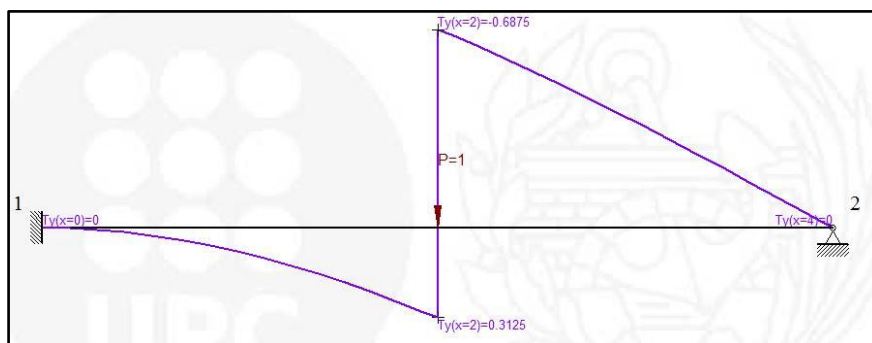


Figura 41. Línea de influencia del cortante en la sección P

1.4. Viga biempotrada

Ejemplo 4. Viga biempotrada de 7 metros con sección de estudio a 4 metros del primer apoyo

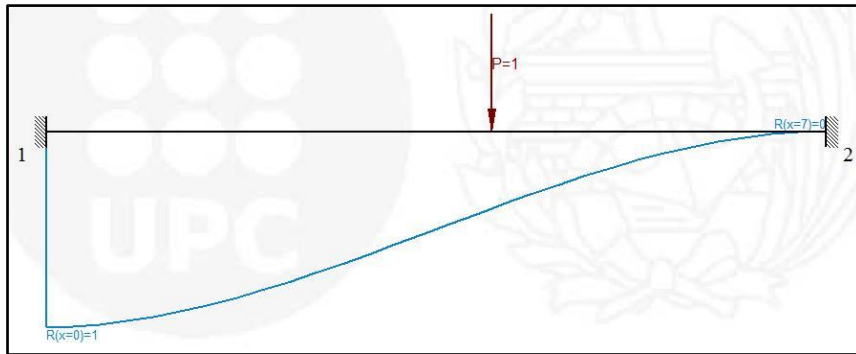


Figura 42. Línea de influencia de la reacción en el apoyo 1

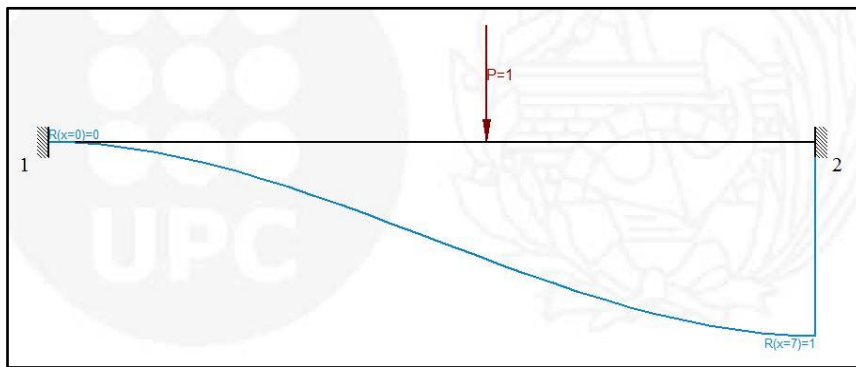


Figura 43. Línea de influencia de la reacción en el apoyo 2

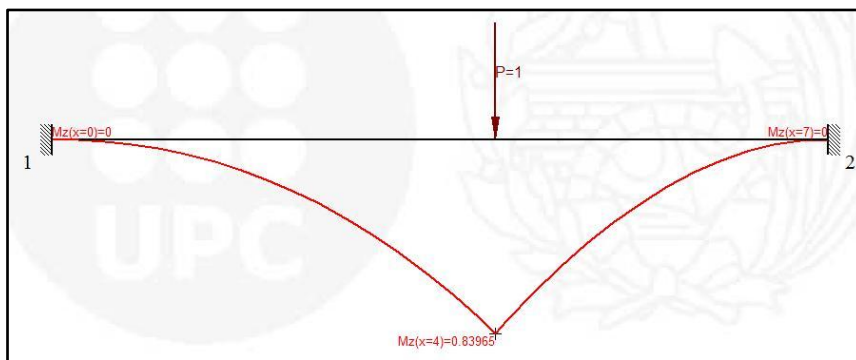


Figura 44. Línea de influencia del momento flector en la sección P

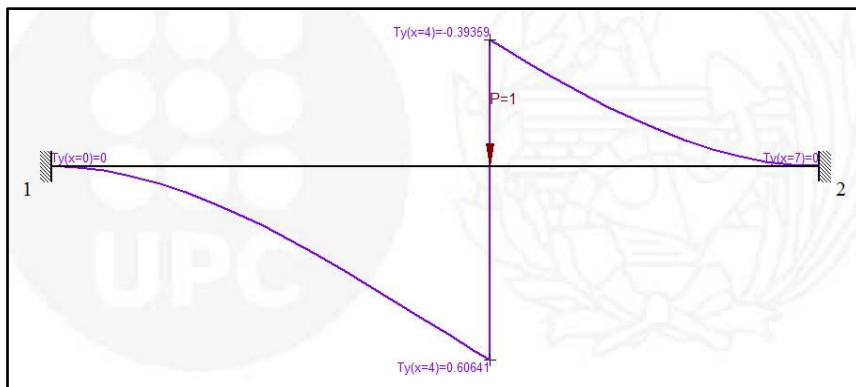


Figura 45. Línea de influencia del cortante en la sección P

1.5. Viga con rótula

Ejemplo 5. Viga con rótula en el nodo 3 con sección de estudio en P

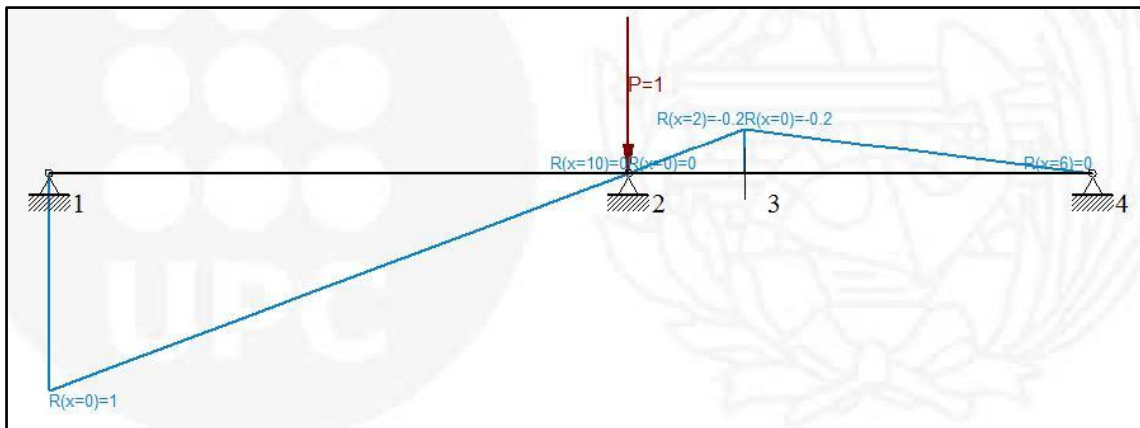


Figura 46. Línea de influencia de la reacción en los el apoyo 1

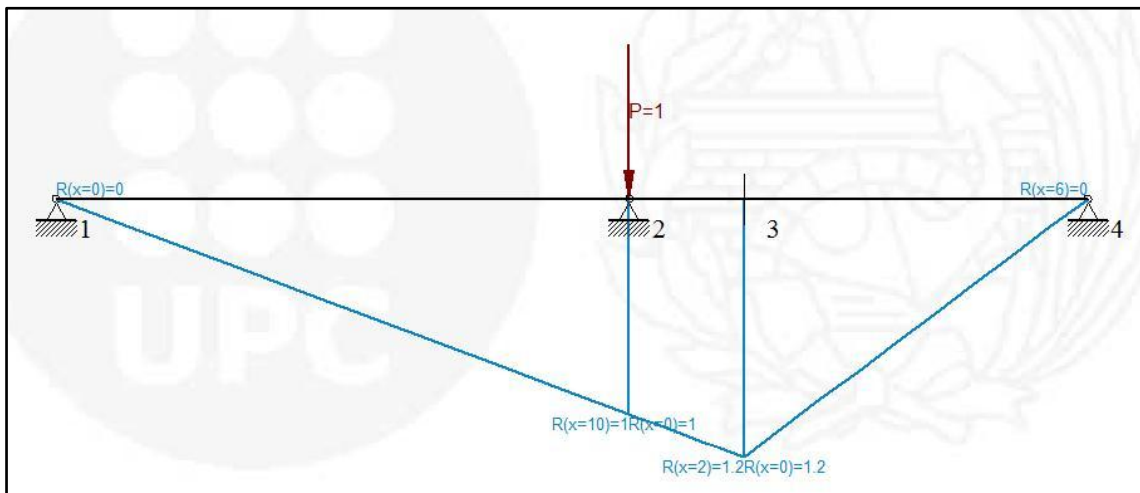


Figura 47. Línea de influencia de la reacción en los el apoyo 2

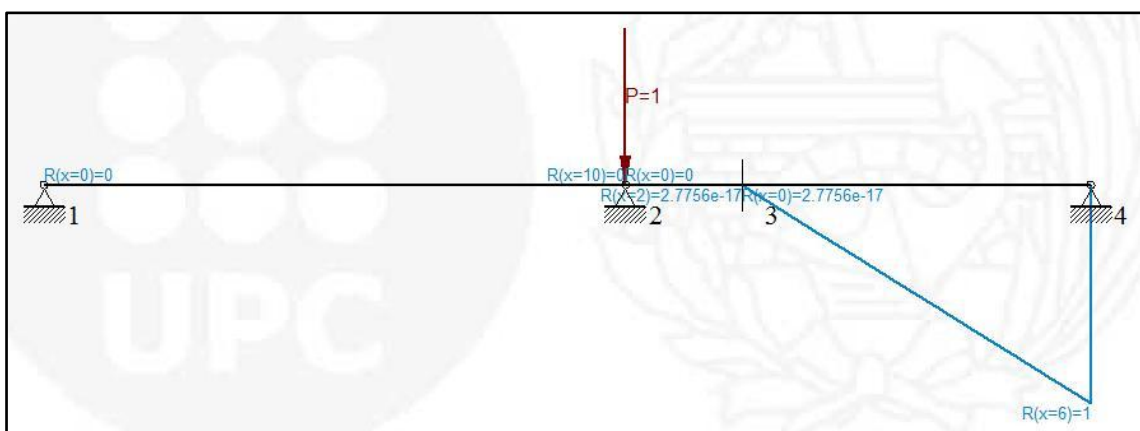


Figura 48. Línea de influencia de la reacción en los el apoyo 4

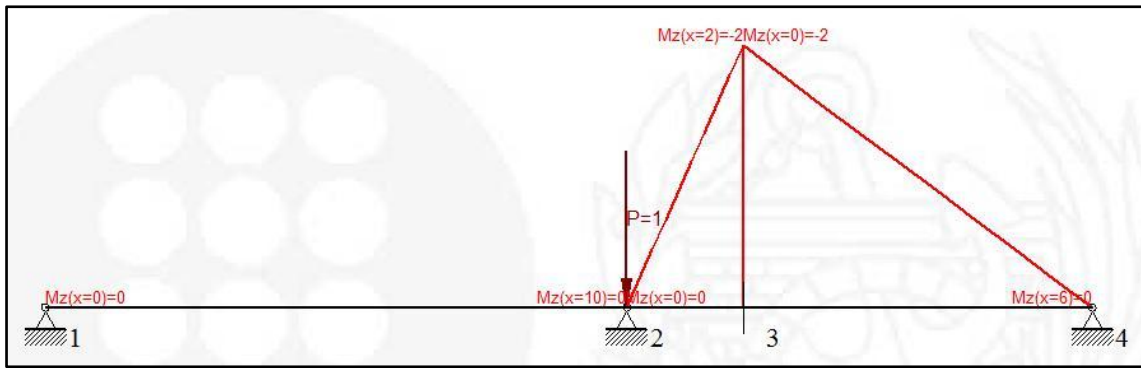


Figura 49. Línea de influencia del momento flector en la sección P (Apoyo 2)

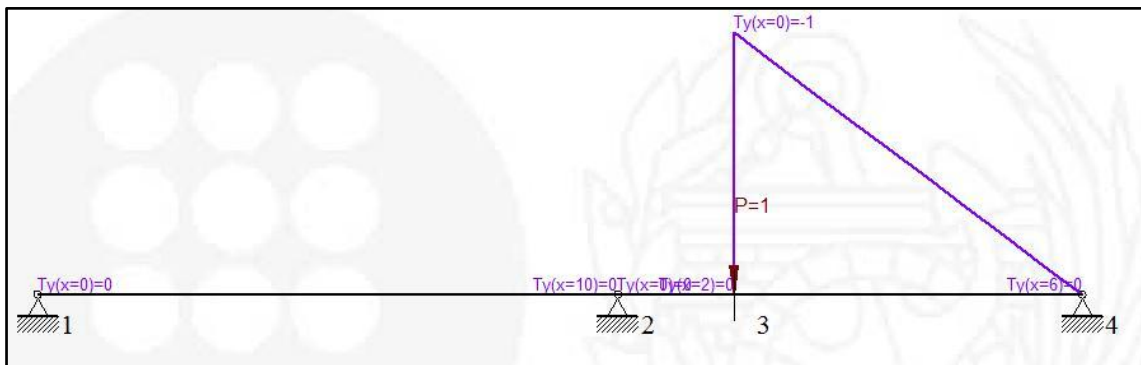


Figura 50. Línea de influencia del cortante en la sección P (Rótula 3)

1.6. Viga isostática de 3 vanos con 2 rótulas

Ejemplo 6. Viga de 3 vanos desiguales con 2 rótulas. Sección a 0.5 metros de la primera rótula

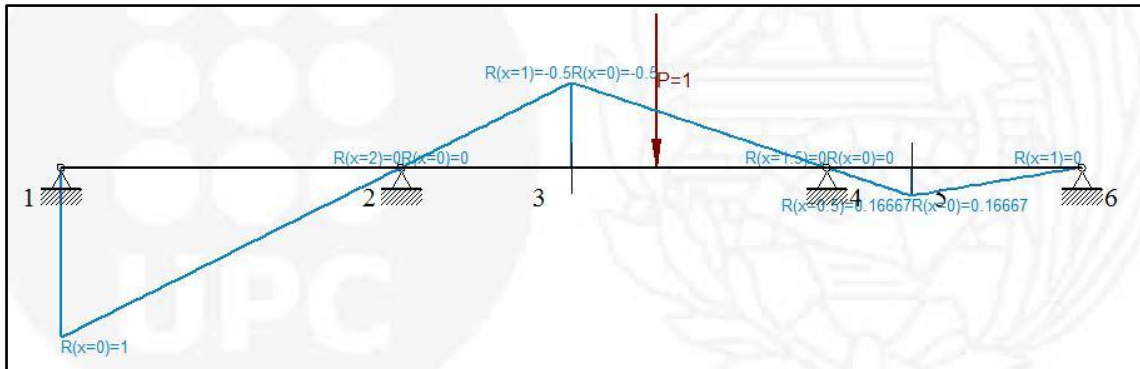


Figura 51. Línea de influencia de la reacción en los el apoyo 1

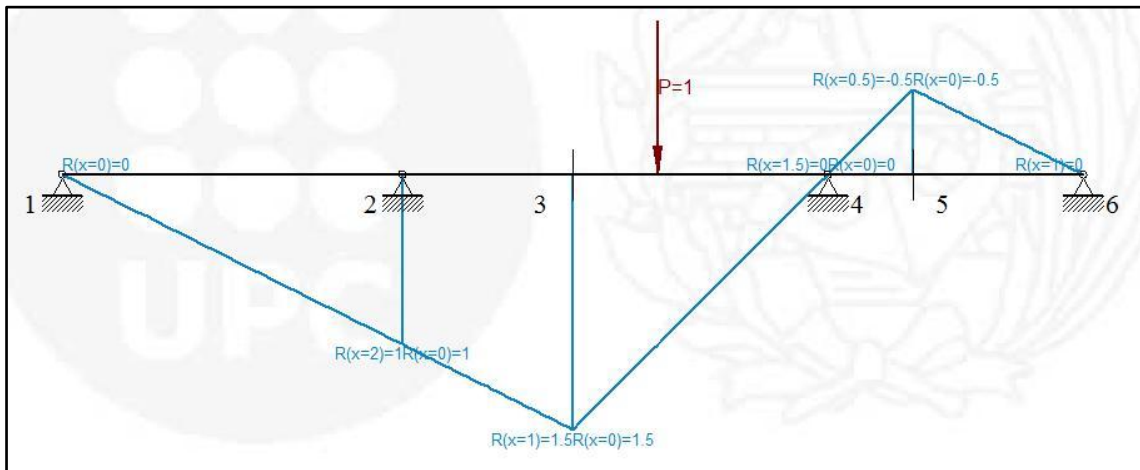


Figura 52. Línea de influencia de la reacción en los el apoyo 2

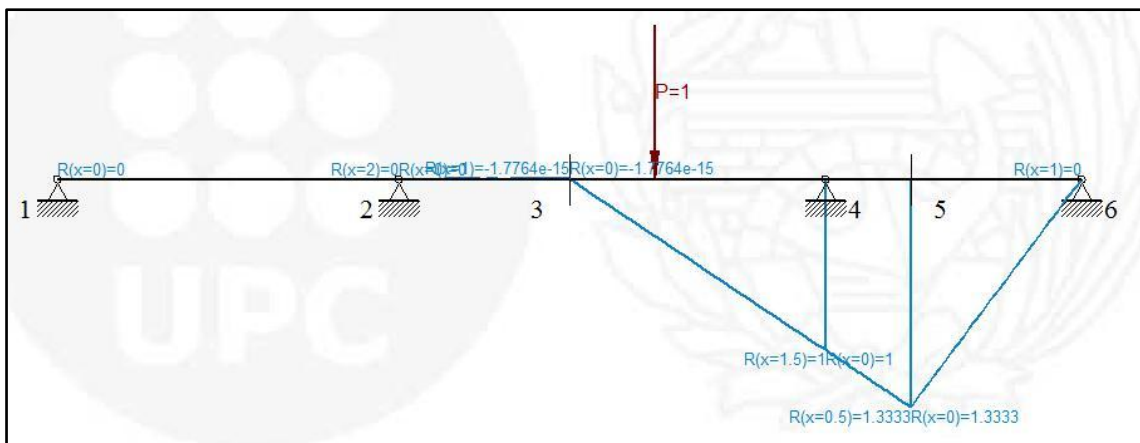


Figura 53. Línea de influencia de la reacción en los el apoyo 4

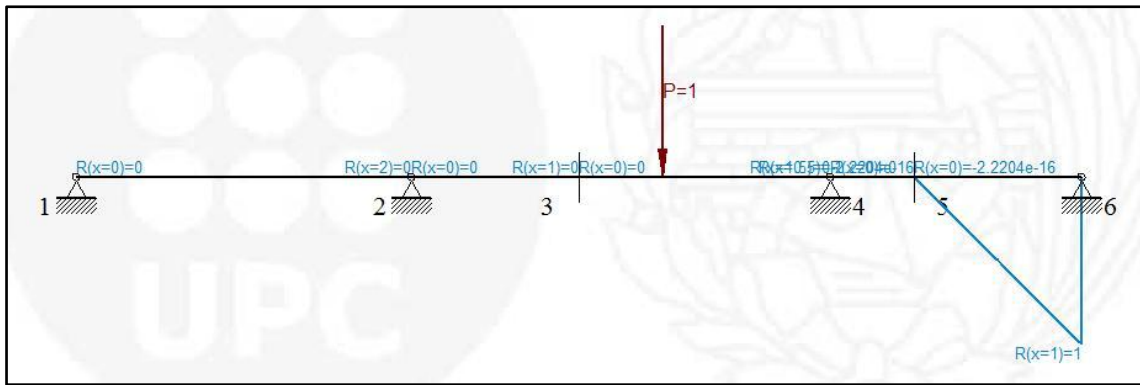


Figura 54. Línea de influencia de la reacción en los el apoyo 6

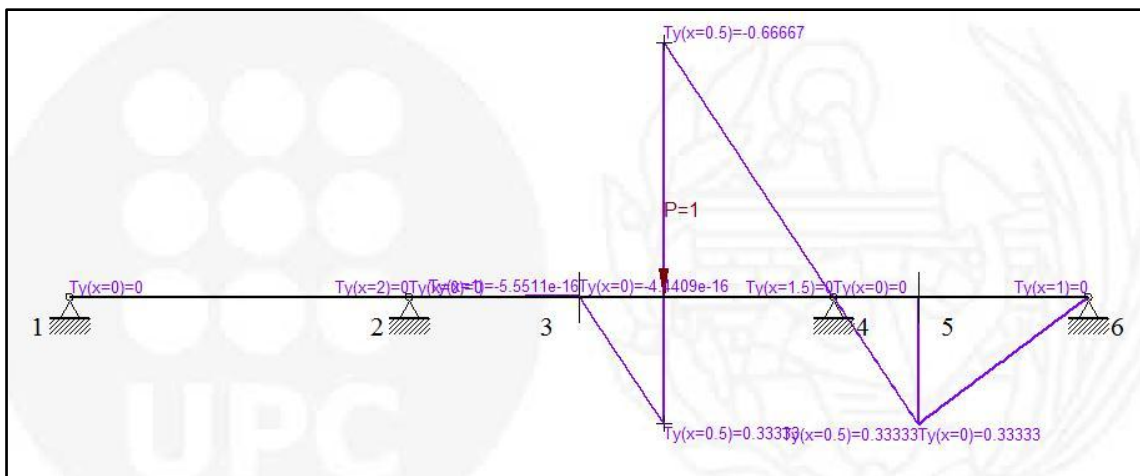


Figura 55. Línea de influencia del cortante en la sección P

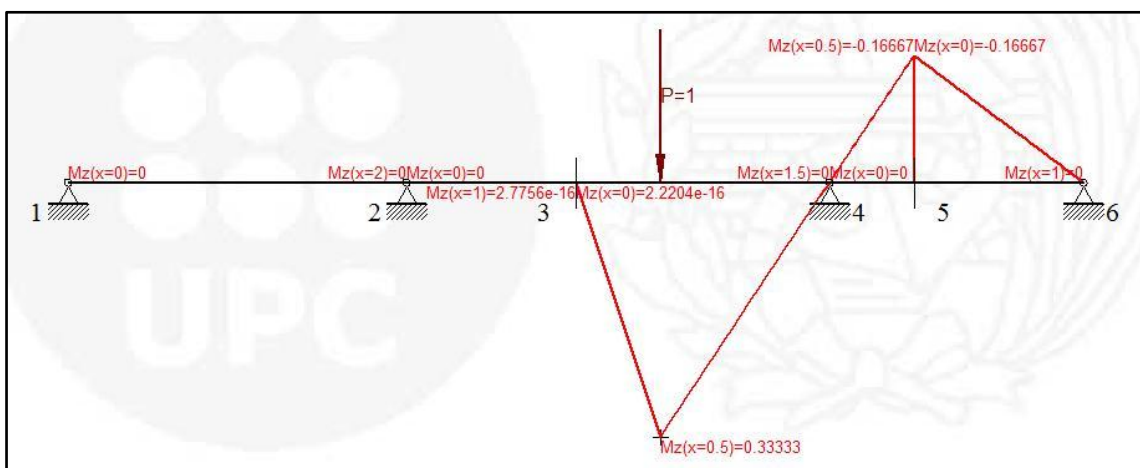


Figura 56. Línea de influencia del momento flector en la sección P

1.7. Viga de 4 vanos con voladizo y rótula

Ejemplo 7. Viga de 4 vanos desiguales y voladizo con sección de estudio en el tercer apoyo

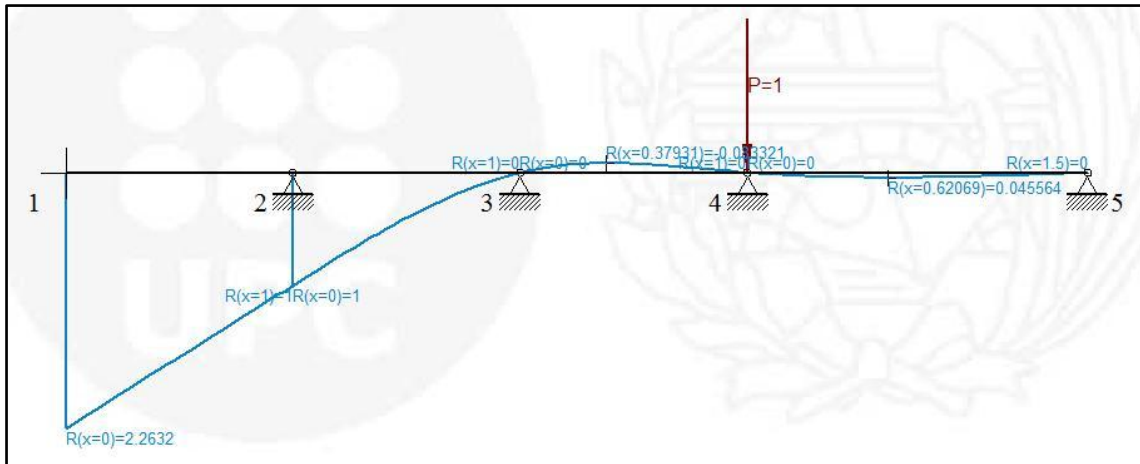


Figura 57. Línea de influencia de la reacción en los el apoyo 2

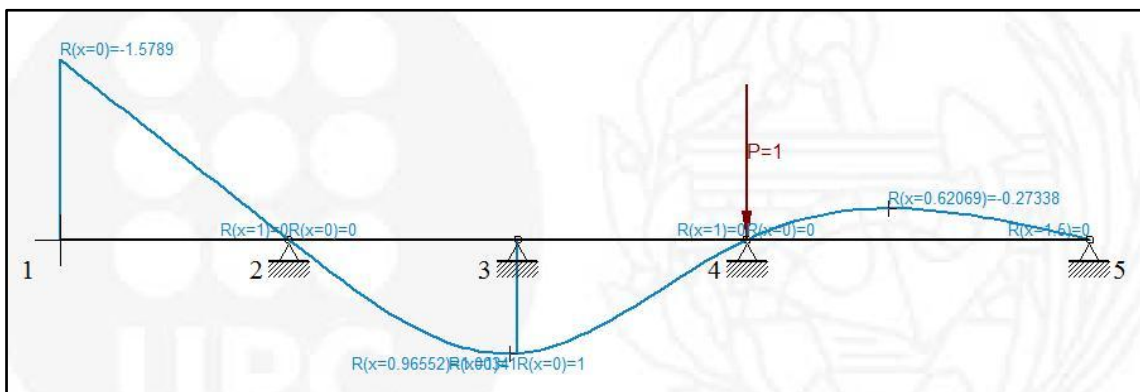


Figura 58. Línea de influencia de la reacción en los el apoyo 3

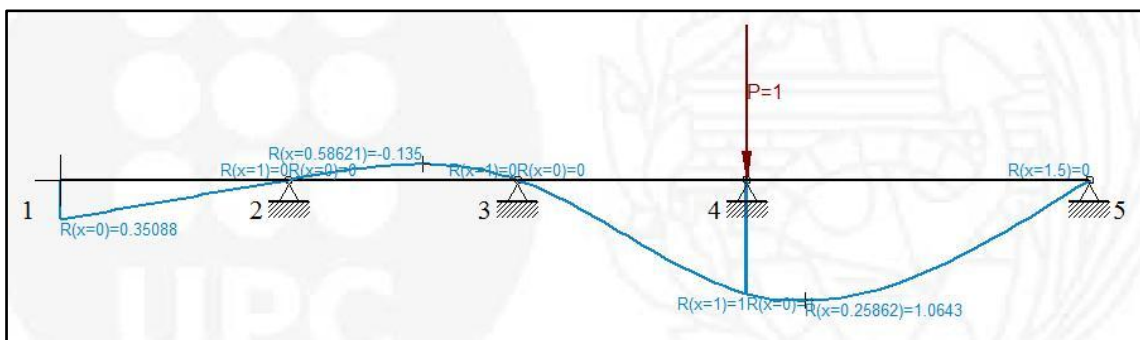


Figura 59. Línea de influencia de la reacción en los el apoyo 4

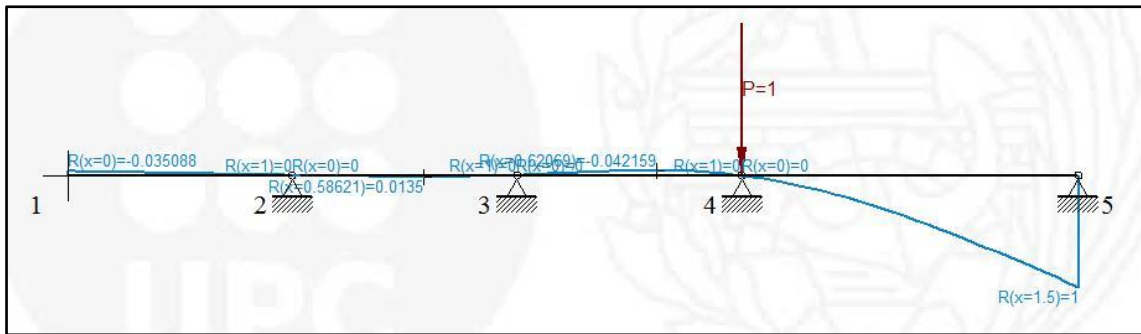


Figura 60. Línea de influencia de la reacción en los el apoyo 5

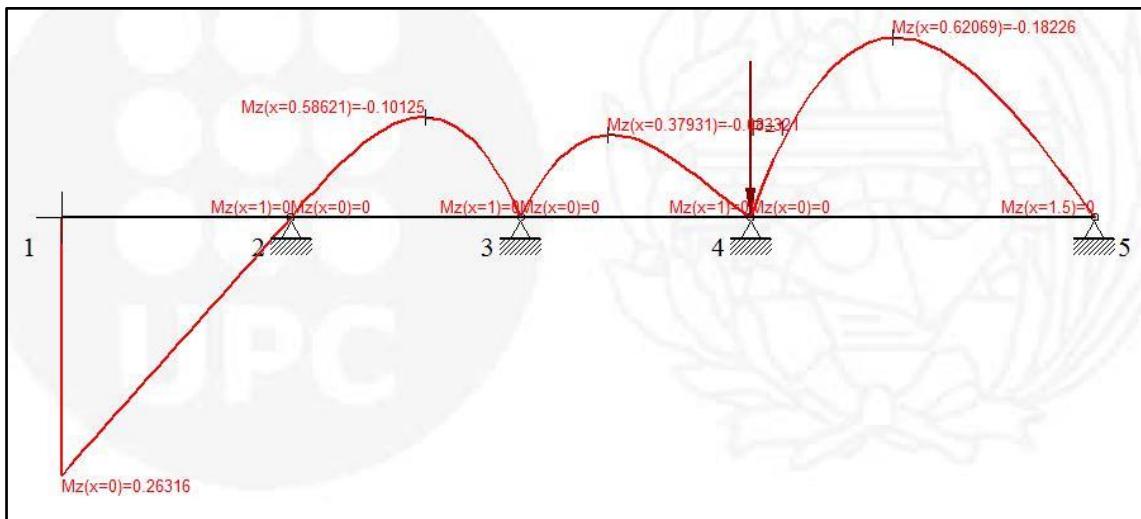


Figura 61. Línea de influencia del momento flector en la sección P

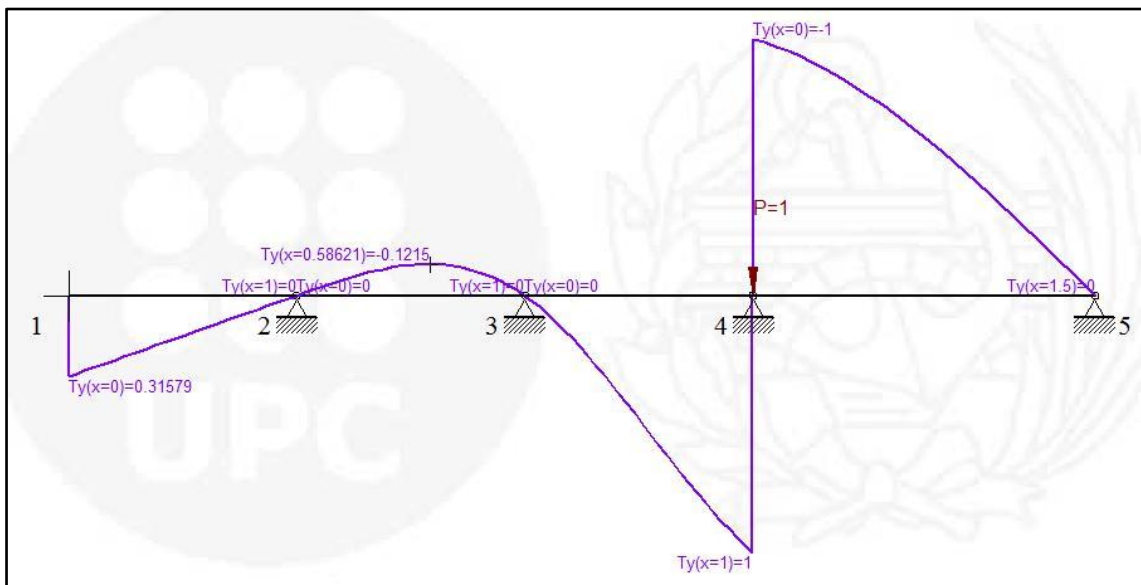


Figura 62. Línea de influencia del cortante en la sección P

2. Conclusiones

Con el desarrollo de este trabajo de fin de grado se ha conseguido el objetivo principal: la creación de un módulo para calcular líneas de influencia de vigas isostáticas e hiperestáticas con el fin de complementar la docencia de las asignaturas de estructuras.

Las líneas de influencia han sido calculadas aplicando la estricta definición de línea de influencia, es decir, una carga puntual y unitaria que va recorriendo la estructura y se va observando qué sucede en la sección de estudio. Por lo tanto, el resultado final ha sido un módulo sencillo, fiable, útil, visual y cómodo de usar aunque no se tenga nociones de Matlab.

Aunque el programa calcula las líneas de influencia para unas determinadas vigas en los ejes X e Y, se podrían implementar otras mejoras que no se han considerado necesarias como las líneas de influencia para axiles o en otros ejes distintos a los nombrados anteriormente. El motivo por el cual no se han calculado también, es que la mayoría de problemas con los que nos vamos a encontrar estarán relacionados con estructuras del tipo grúas móviles, puentes que soportan el tráfico de camiones o de trenes, etc.

Por otro lado, el trabajo entrega un programa completo, en general, para complementar la docencia pero parcialmente completo en lo referente a líneas de influencia de estructuras, lo que permitirá que se pueda seguir mejorando el módulo por futuros desarrolladores. Se ha creído conveniente enumerar algunas carencias o aspectos mejorables del programa según el grado de dificultad.

Respecto a las posibles mejoras relativamente fáciles de implementar se podrían considerar:

- Líneas de influencia en todos los ejes y no sólo las líneas de influencia de las reacciones, del cortante en el eje Y y del momento flector en el eje Z.
- Calcular las envolventes de las líneas de influencia del momento flector.
- Decidir cuántos puntos equidistantes queremos tener en la ventana de inicio y no por defecto.
- Optimizar el código sabiendo que las líneas de influencia de las vigas isostáticas son rectas.

Respecto a las posibles mejoras relativamente difíciles de implementar se podrían considerar:

- Programarlo de forma que al clickar en el dibujo salga el valor de cualquier ordenada. Dado que las líneas de influencia para vigas no isostáticas no son rectas, encontrar una función que se adapte es complicado. Se podrían usar mínimos cuadrados para encontrar una función que se adapte bien, tal como se estudia en la asignatura de Métodos Numéricos aunque es complicado dado que las líneas no siguen ninguna función determinada.
- Pasar de vigas horizontales a vigas inclinadas, pórticos o vigas articuladas.
- Ampliar el módulo a trenes de carga.

Como conclusión final, podemos decir que los objetivos principales del trabajo se han llevado a cabo: crear un módulo que calcule las líneas de influencia de manera rápida para cualquier tipo de viga no inclinada. Me habría gustado poder haber hecho algunas de las mejoras expuestas anteriormente, pero mi nivel de Matlab era muy escaso comparado con el nivel de los diseñadores iniciales del programa. Por ese motivo, ha sido bastante complicado hacer el módulo aunque gracias a él, he podido mejorar mis conocimientos sobre Matlab más de lo que había aprendido durante toda la carrera.

Este trabajo además de ser un Trabajo de Final de Grado también puede servir de ayuda para futuros desarrolladores que quieran optimizar el módulo de las líneas de influencia dado que, en estructuras, sobre todo en puentes, es una de las cosas más importantes a tener en cuenta y no se les da la suficiente importancia que se les debería dar.

3. Bibliografía

- [1] Blanco Díaz, E. y Cervera Ruiz, M. (2014). *Apuntes Líneas de Influencia*. Escuela Técnica Superior de Ingenieros Caminos, Canales y Puertos, Universidad Politécnica de Catalunya. Barcelona

- [2] García de Jalón, J., Rodríguez, J.I. y Vidal, J. (2005). *Aprenda Matlab 7.0 como si estuviera en primero*. Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid. Diciembre, Madrid.

- [3] Páginas de internet:

<http://www.ing.unlp.edu.ar/estruc3b/liin.pdf>

http://estructuras.eia.edu.co/estructurasl/lineas%20de%20influencia/l%C3%ADneas_de_influencia.htm

https://www.u-cursos.cl/usuario/e32608c194c5b4806b5c633a33da4dd3/mi_blog/r/Lineas_de_Influencia-Primera_Parte.pdf

- [4] The MathWorks, Inc. (1994-2015). *Academia*. Natick, US. Recuperado de <http://www.mathworks.com>.